

AFFDL-TR-76-59

(12)

DT

ADA 029021

COMPASS COPE FLIGHT CONTROL SYSTEM REDUNDANCY STUDY

ROCKWELL INTERNATIONAL
COLLINS RADIO GROUP
CEDAR RAPIDS, IOWA

8 APRIL 1976

TECHNICAL REPORT AFFDL-TR-76-59
FINAL REPORT SEPTEMBER - DECEMBER 1975

Approved for public release; distribution unlimited

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

DDC
REFINED
AUG 30 1976
B

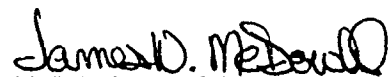
g

NOTICE

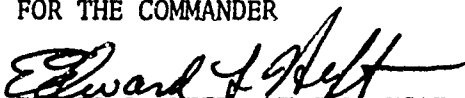
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical memorandum has been reviewed and is approved for publication.


JAMES W. MCDOWELL
Project Engineer

FOR THE COMMANDER


EDWARD L. HEFT, LT COL, USAF
Chief, Terminal Area Control &
Flight Control Division

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFFDL-TR-76-59	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Compass Cope Flight Control System Redundancy Study.		5. TYPE OF REPORT & PERIOD COVERED Final Report Sep-Dec 75 Revised 8 April 76
7. AUTHOR(s) R. F. Tribuno, J. A. Klein, D. R. Stover K. G. Martin		6. PERFORMING ORG. REPORT NUMBER S-76-2
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rockwell International Collins Radio Group Cedar Rapids, Iowa		8. CONTRACT OR GRANT NUMBER(s) F33615-73-C-3051
11. CONTROLLING OFFICE NAME AND ADDRESS Calspan Corporation Buffalo, New York		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 19575001
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson AFB, Ohio 45433		12. REPORT DATE 8 Apr 1976
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release, Distribution Unlimited		13. NUMBER OF PAGES 113
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15. SECURITY CLASS. (of this report) Unclassified
18. SUPPLEMENTARY NOTES		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Compass Cope RPV Digital Flight Control System Redundancy and Monitoring		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report presents the result of study conducted by Collins Radio Co, under contract to Calspan Corporation, to define and discuss flight control system reliability requirements, monitoring techniques and fault analysis. Also included are hardware and software reliability, implementation tradeoff discussions, configuration redundancy candidates and recommendations for further study.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 904

FOREWORD

From September through December 1975, Collins Radio Co studied several configuration candidates of digital flight control systems for the Compass Cope Remotely Piloted Vehicle. This study addressed the problem by establishing a set of Compass Cope reliability requirements and comparing these requirements to derived reliabilities along with system costs for several different system redundancy configurations. This report also discusses monitoring techniques, fault analysis, hardware and software reliability and implementation tradeoffs.

The study was conducted with the authority of the Remotely Piloted Vehicles Autoland System Study, project/task number 19575001.

ACCESSION	
WTM	White Section <input checked="" type="checkbox"/>
BDC	Red Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. CODE/SPECIAL
A	

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.0	INTRODUCTION	1
2.0	SUMMARY	2
3.0	RELIABILITY REQUIREMENTS	6
3.1	Definition of Probabilities	6
3.2	Compass Cope/Manned-Fighter Analogy	6
3.3	Vehicle Loss Requirement	8
3.4	Takeoff and Recovery Loss Requirement	9
3.5	Mission Abort Requirement	10
3.6	Civil Airspace Hazards	12
3.6.1	Ground Hazards	12
3.6.2	Midairs	12
3.7	Summary of Reliability Requirements	13
4.0	STUDY GROUND RULES & ASSUMPTIONS	14
4.1	Flight Profile Model	14
4.2	Operational Assumptions	14
4.3	Performance Assumptions	14
4.4	Study Ground Rules	16
5.0	FAULT ANALYSIS APPROACH	18
5.1	Fault Analysis Model	19
5.2	Vehicle Loss Fault Tree	19
5.3	Mission Abort Fault Tree	24
6.0	ANALOG VS. DIGITAL HARDWARE	26
6.1	Flight Computer	26
6.1.1	System Size/Weight/Cost Reductions	26
6.1.2	Inherent Advantages of Digital Technology	26
7.0	DIGITAL PROCESSOR RELIABILITY & MONITORING	30
7.1	General	30
7.2	Self-Monitoring Methods	32
7.2.1	CPU Tests	32
7.2.2	Memory Tests	32
7.2.3	I/O Tests	33
7.2.4	Software and System Checks	33
7.3	Processor Hardware Reliability	34
7.4	Software Reliability	40
7.5	Use of a Microprocessor as a Monitor	41
8.0	SYSTEM CONFIGURATION CANDIDATES	43
8.1	Configuration Components	43
8.1.1	Sensors	44
8.1.2	Flight Computers	44
8.1.2.1	The Conventionally-Monitored Computer	47
8.1.2.2	The Highly-Monitored Computer	47

TABLE OF CONTENTS (Continued)

<u>Section</u>		<u>Page</u>
8.1.3	Servos	50
8.1.4	Sensor Interface	50
8.1.5	Servo Interface & Equalization	53
8.1.6	Independent Redline Monitor	53
8.1.7	Backup Analog Flight Computer	53
8.2	Definition of Candidates	56
8.2.1	Configuration A	56
8.2.2	Configuration B	56
8.2.3	Configuration C	56
8.2.4	Configuration D	59
8.2.5	Configuration E	63
9.0	FAULT ANALYSIS RESULTS	65
9.1	Fault Analysis Summary	65
9.2	FCS Probability-of-Loss Discussion	65
9.3	Inline Monitoring of a Digital Processor	68
9.4	Inline vs. Redline Monitoring	68
9.5	Effects of Software Reliability	68
9.6	Probability-of-Mission-Abort Discussion	69
10.0	RECOMMENDATIONS FOR FURTHER STUDY	70
11.0	REFERENCES	77
APPENDICIES		
A	SENSOR & SERVO REDUNDANCY REQUIREMENTS	74
B	DETAILED FAULT ANALYSIS	87
C	DERIVATION OF NON-RECURRING PLANNING ESTIMATES	106
D	AIRCRAFT LOSS DATA, 1973	109

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
3.1	Compass Cope/Manned-Fighter Analogy	7
4.1	Flight Profile Model	15
5.1	Fault Analysis Model	20
5.2	Vehicle Loss Fault Tree	21
5.3	Vehicle Loss in Cruise Fault Tree	21
5.4	Vehicle Loss in Takeoff Fault Tree	22
5.5	Vehicle Loss in Recovery Fault Tree	23
5.6	Mission Abort Fault Tree	25
6.1	Digital Cost Comparison, Fail Operative Flight Control System	28
7.1	Digital Processor Reliability Model	31
7.2	Incremental Cost of Inline Monitoring	38
7.3	Microprocessor Monitor	42
8.1	Candidate Shipset	46
8.2	Conventionally-Monitored Digital Flight Computer	48
8.3	Highly-Monitored Digital Flight Computer	49
8.4	Dual Servo Configuration with Alternate-Engage Switching	51
8.5	Dual Servo Configuration with Torque Summing	52
8.6	Redline Monitor Functional Diagram	54
8.7	Backup Analog Flight Computer	57
8.8	Configuration A	58
8.9	Configuration B	60
8.10	Configuration C	61
8.11	Configuration D	62
8.12	Configuration E	64

LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
A-1	Servo Interface, Command Switching vs. Voting, Triple Computers	85
A-2	Servo Interface, Dual Computers	86
B-1	Fault Tree, Config. B, Vehicle Loss Cruise	89
B-2	Fault Tree, Config. B, Vehicle Loss Takeoff	90
B-3	Fault Tree, Config. B, Vehicle Loss Recovery	91
B-4	Fault Tree, Config. C, Vehicle Loss Cruise	92
B-5	Fault Tree, Config. C, Vehicle Loss Takeoff	93
B-6	Fault Tree, Config. C, Vehicle Loss Recovery	94
B-7	Fault Tree, Config. D, Vehicle Loss Cruise	99
B-8	Fault Tree, Config. D, Vehicle Loss Takeoff	100
B-9	Fault Tree, Config. D, Vehicle Loss Recovery	101
B-10	Fault Tree, Config. E, Vehicle Loss Cruise	102
B-11	Fault Tree, Config. E, Vehicle Loss Takeoff	103
B-12	Fault Tree, Config. E, Vehicle Loss Recovery	104
B-13	Fault Tree, Mission Abort	105

LIST OF TABLES

<u>Table</u>		<u>Page</u>
S-1	System Cost/Reliability Comparison, Ideal Software Assumed	4
3.1	Recovery & Takeoff Accidents, 1973	11
3.2	Suggested Reliability Requirements	13
6.1	Analog vs. Digital Comparison for Redundant Air Transport Flight Control System	27
7.1	Self-Checking Technique Effectiveness, CAPS-4 Processor	35
7.2	Incremental Cost of Self-Checking Techniques, CAPS-4 Processor	36
7.3	Estimating Level of DFC Self Monitoring	37
8.1	Representative Compass Cope Flight Control System Equipment	45
9.1	System Cost/Reliability Comparison, Ideal Software Assumed	66
9.2	System Cost/Reliability Comparison with Unreliable Software	67

1.0 INTRODUCTION

The peculiar operational requirements of the Compass Cope RPV impose stringent reliability requirements on the vehicle flight control system. These operational requirements include a mission duration of over 24 hours, the capability of fully automatic flight from takeoff through recovery phases, and the capability of operating within civil airspace and into civil airfields.

The Compass Cope development program requires a cost effective FCS configuration definition capable of satisfying the appropriate reliability requirements. The Air Force Flight Dynamics Laboratory responded to this need by administering the study effort reported by this document.

This study addresses the problem by establishing a set of Compass Cope reliability requirements and comparing these requirements to derived reliabilities along with system costs for several different system redundancy configurations. The study is comprised of the following tasks:

1. Definition of FCS reliability requirements
2. Definition of study ground rules and assumptions
3. Definition of the study fault analysis approach
4. Flight computer implementation tradeoff study
5. Study of digital processor reliability, monitoring techniques, and software reliability
6. Definition and derivation of configuration redundancy candidates
7. Discussion of fault analysis results and system tradeoffs
8. Recommendations for further study

Each of the above tasks constitutes a section of this report.

2.0 SUMMARY

A suggested set of reliability requirements for the Compass Cope Flight Control System and data link are established. The requirements are defined in terms of probability of vehicle loss during takeoff, recovery, and over the entire flight. A probability of mission abort is also included. The "target" requirements, based on fighter-aircraft statistics and derived for a range of system maturity levels, are summarized below.

PROGRAM MATURITY LEVEL	SUGGESTED FCS PROBABILITY			
	VEHICLE LOSS			MISSION ABORT
	ENTIRE FLIGHT	RECOVERY	TAKEOFF	
NEW (F111)	0.0033	41×10^{-6}	25×10^{-6}	0.072
MATURE (F4)	0.0017	21×10^{-6}	13×10^{-6}	0.032

TABLE OF SUGGESTED
RELIABILITY REQUIREMENTS

A tradeoff study was performed to establish the optimum mix of analog and digital hardware. Experience strongly suggests that the flight computers be digital processors to minimize space, weight, and cost and to achieve a thorough self-test capability.

The reliability of digital processors and software is explored. Tradeoffs of in-line monitoring levels vs. cost are established. Both undetected processor hardware failures and software algorithm problems are potentially hazardous in a flight control system. Software problems, though very difficult to quantify, can occur and generally defy detection by in-line monitoring. High levels of hardware monitoring, however, may be achieved at modest extra cost. For those applications that require a well-monitored processor, it is recommended that an independent microprocessor, packaged within the flight computer, perform the in-line monitor function. The same microprocessor can then perform an independent red-line monitor and backup autopilot-computer function.

Five system configurations, each with a fail-operational set of sensors and servos, but with different flight computer redundancy and monitoring, were evaluated:

SYSTEM	COMPUTER REDUNDANCY
A	Single conventionally-monitored digital flight computer with separate backup analog autopilot computer.
B	Same as A with independent red-line monitor.
C	Dual conventionally-monitored digital flight computers with separate backup analog computer and independent red-line monitor.
D	Dual highly-monitored digital flight computers, each with red-line monitor and backup autopilot functions programmed on an internal microprocessor.
E	Triple conventionally-monitored digital flight computers without red-line monitor or backup autopilot.

Vehicle-loss probabilities were obtained for these system configurations from a fault tree probability analysis. Total system costs and weights were compiled. Component data was obtained for a representative RPV shipset of equipment. To demonstrate the vulnerability of the different configurations to software problems, results were obtained for both ideal (error-free) and unreliable (containing algorithm and other errors) software. System cost/reliability tradeoffs for the ideal software case are summarized in Table S-1 below.

The following conclusions can be extrapolated from the system cost/reliability analysis:

1. Fail-operative sets of critical sensors (triple unmonitored or dual monitored) are required. Crossfeeding of sensor data into the flight computers is required except in the case of dual monitored sensors feeding dual computers.
2. Fail-operative servo configurations are required for all of the flight-critical control surfaces and throttle. Crossfeeding of servo commands from the flight computers into the servos is required for all configurations, though for different reasons.
3. In cruise the loss probabilities of all configurations, except A, are within 5% of 0.002. The loss probability for A is 0.0035.
4. Data link and cruise sensor contributions swamp out flight-computer redundancy and monitoring techniques and cause nearly-identical loss probabilities in cruise.
5. The cruise loss probabilities for all systems exceed the suggested study requirement of 0.0017. However, if high-reliability (2X MTBF improvement) vertical gyros and data links are substituted, the resulting probabilities can be reduced significantly below 0.0017.
6. All configurations comfortably meet the suggested takeoff and recovery requirements.
7. Unlike in cruise, computer redundancy markedly improves system reliability in takeoff and recovery phases.
8. Ranking in takeoff/recovery in order of decreasing reliability - E, D, C, B, A.
9. In cruise, all but A and E can accommodate unreliable software with less than 10% change in reliability.
10. If unreliable software is assumed, the ranking in takeoff/recovery in order of decreasing reliability becomes - D, C, B, E, A.
11. Red-line (performance) monitoring is extremely beneficial and can detect, in most cases, otherwise undetected hardware failures and software problems.
12. No configuration meets the suggested mission-abort requirement of 0.034, though high -reliability sensors yield considerable improvement. A redefinition of the abort groundrules may be in order, given the unusually long mission duration of the Compass Cope.
13. Ranking in order of decreasing system cost - E, C, D, B, A.

SYSTEM CONFIGURATION	COSTS		WEIGHT	FCS PROBABILITY			
				VEHICLE LOSS			MISSION ABORT
	HARDWARE	DESIGN		ENTIRE FLIGHT	TAKEOFF	RECOVERY	
	(K dollars)	(K dollars)	(lbs)	(x 10 ⁻⁶)	(x 10 ⁻⁶)		
A	210	1329	370	0.0035	0.556	4.22	0.133
B	219	1896	378	0.00203	0.556	2.62	0.137
C	279	1896	413	0.00198	1.04	0.888	0.145
D	273	1008	395	0.00193	0.236	0.214	0.139
E	323	958	430	0.00203	0.05	0.059	0.144

SYSTEM COST/RELIABILITY COMPARISON

IDEAL SOFTWARE ASSUMED

TABLE S-1

If a recommendation were to be made, it would be between configuration D and B, depending on the emphasis placed on recovery reliability. Both configurations have nearly identical cruise loss probabilities. Though costing 25% more than B, D exhibits an order-of-magnitude better recovery reliability.

3.0 RELIABILITY REQUIREMENTS

3.1 Definition of Probabilities

It is reasonable to specify the Compass Cope flight control system (FCS) reliability requirements in terms of the following vehicle loss and mission abort probabilities:

- Probability of Vehicle Loss During Takeoff/Recovery - The probability of losing the vehicle due to a FCS malfunction during the takeoff and recovery phases.
- Probability of Vehicle Loss - The probability of losing the vehicle due to a FCS malfunction anywhere along the entire flight profile, including takeoff, recovery, and cruise phases.
- Probability of Mission Abort - The probability of aborting a mission, though not necessarily losing the vehicle, due to a FCS malfunction. An abort is assumed to occur when nominal system redundancy is degraded by a first failure within the FCS.

The probability of vehicle loss specifies the probability of losing the RPV throughout a typical 24-hour mission. It relates directly to yearly RPV attrition costs. As would be expected, the cruise loss contribution predominates.

The probability of vehicle loss during recovery and takeoff phases gives high resolution to these critical maneuvers. Though this parameter only concerns irreparable damage to the RPV, and not damage to property or personnel on the ground, a method for assessing these other hazards is discussed below.

The classical way to improve system performance reliability is to add redundancy, usually at the expense of system reliability. The mission-abort probability provides a measure of overall system reliability.

3.2 Compass Cope/Manned-Fighter Analogy

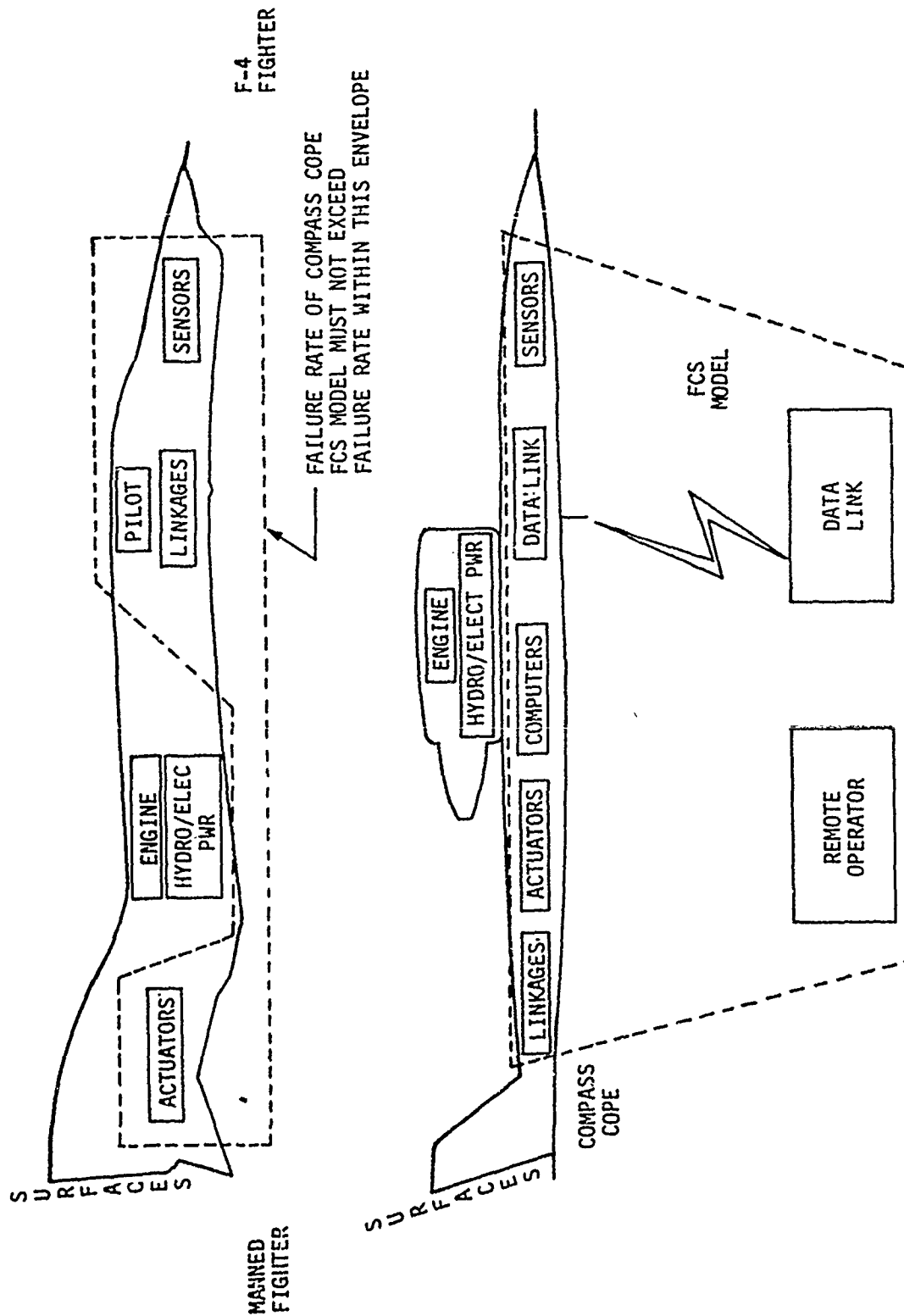
If the Compass Cope can be shown to be as safe as manned military aircraft when flying within civil airspace and into civil airfields, Cope will, most likely, obtain operational approval of the aviation community, since these aircraft have already been accepted. Values for the suggested Cope loss probabilities can be derived from military aircraft loss statistics. Various redundancy configurations for the Cope FCS can then be evaluated against these suggested values.

It is difficult to find a military aircraft similar to Cope in either performance, physical characteristics or mission requirements. However, an argument can be made for using fighter aircraft over other military types as a reference. The analogy is depicted in Fig. 3.1.

Considering the RO the remote pilot of the RPV, both Cope and often fighter aircraft have one pilot. Additionally, the benign environment of the RPV can be traded for the benefits of an on-board pilot in the fighter.

COMPASS COPE FCS MODEL/MANNED-FIGHTER ANALOGY

FIG. 3.1



Arguments can be made against using other military aircraft types and drones, for that matter, as a reference.

- Statistics available for utility aircraft suffer from an insufficient data base.
- Statistics available for drones and RPV's suffer from an unperfected takeoff/recovery science and RO error.

3.3 Vehicle Loss Requirement

A suggested target value for probability of vehicle loss can be derived from loss statistics available in USAF TAC reports¹ for the five-year period 1966 to 1970. In accordance with the analogy of Fig. 3.1,

Fighter loss rate attributable to analogous Cope FCS failures

= Loss rate attributable to pilot

+ Loss rate attributable to sensors

+ Loss rate attributable to fighter FCS

The loss rate for TAC fighters for all causes² between 1966 and 1970 is

$$\frac{1}{8300} = 120 \times 10^{-6} \text{ hr}^{-1}$$

The data shows that 40% of these losses are attributable to pilot error.
Thus

$$\text{Loss rate attributable to pilot} = .4 \times 120 \times 10^{-6} = 48 \times 10^{-6}$$

1. Ref. 8, pp. 63, 64

2. Excludes contributions from electrical and hydraulic power failures.

7-7-1

If it is assumed that sensors, maintenance, and miscellaneous causes account for another 15%, then, similarly,

Loss rate attributable to sensors, etc. $= 18 \times 10^{-6}$

In the same time period,

F-111 loss rate attributable to FCS $= 70.6 \times 10^{-6}$

F-4 loss rate attributable to FCS $= 5.8 \times 10^{-6}$

Combining,

Fighter loss rate attributable to analogous Cope FCS failures

$$\begin{cases} 136.6 \times 10^{-6} & \text{F-111} \\ 71.8 \times 10^{-6} & \text{F-4} \end{cases}$$

The probability of vehicle loss on a 24-hour mission then becomes:

0.00328 for F-111

and

0.00172 for F-4

The 2-to-1 disparity between the two aircraft is significant and relates to the different maturity levels. The F-4 was considered significantly more mature than the F-111 in the 1966-70 time period. Thus it is reasonable to impose initially a higher loss probability on the Compass Cope FCS than what might be the ultimate target probability for a mature program.

3.4 Takeoff and Recovery Loss Requirements

Similarly, suggested target probabilities may be derived for both recovery and takeoff phases from the expression:

Probability of Vehicle Loss During Recovery/Takeoff

$$= (\text{Aircraft Loss Rate, All Causes}) (\text{Average Flight Duration}) \\ (\text{Fraction of Accidents Occurring During Recovery/Takeoff}) \\ (\text{Fraction of Accidents Attributable to Pilot + PFCS + Sensors})$$

Table 3.1 gives the raw data and vehicle loss probability for general aviation, U.S. air carriers, and AF cargo and fighter aircraft for the year 1973. Average flight durations were estimated. The fraction of accidents (0.6) attributable to pilot, PFCS, or sensors was adopted from section 3.3 above and is comprised of:

Pilot	40%
PFCS	5%
Sensors,	
etc.	<u>15%</u>
Total	60%

It may be observed that fighters have the highest recovery and takeoff loss rates and cargo and carriers the lowest. It should be stated that the U.S. carrier data is based on a small sample space of only a total of 7 losses from all causes for 1973. As might be expected, military cargo and U.S. carrier aircraft have similar recovery loss rates. Takeoff loss rates, however, are dissimilar. General aviation loss rates are in between. Because benefit cannot be derived from an on-board pilot, it is probably most realistic to select the highest loss-rate probabilities for the Compass Cope targets. These correspond to the fighter values of 21×10^{-6} and 13×10^{-6} for recovery and takeoff phases, respectively.

3.5 Mission Abort Requirement

Once Compass Cope technology reaches maturity, mission-abort rates will be primarily a function of overall system reliability. As discussed below in Section 4.0, it is assumed that a mission will be aborted and the RPV turned back toward home whenever a failure causes a degradation of the nominal FCS redundancy.

In keeping with the manned-fighter analogy, it is reasonable to use F-111 statistics for the mission abort requirement because of its relatively level of sophistication. For the F-111

$$\text{In-Flight Abort Rate}^1 = 134 \times 10^{-5} \text{ hr}^{-1}$$

1. Ref. 8, p. 87

<u>AIRCRAFT TYPE</u>	<u>A/C LOSS RATE, ALL CAUSES</u>	<u>ESTIMATED AVG FLT DURATION</u>	<u>FRACTION RECOVERY/TAKEOFF ACCIDENTS</u>	<u>FRACTION PILOT + PFCS + SENSORS</u>	<u>PROBABILITY OF A/C LOSS</u>
US AIR CARRIERS ¹ ALL OPERATIONS	1.08 x 10 ⁻⁶	3.hrs	.71/.14	.6	1.4 x 10 ⁻⁶ /.27 x 10 ⁻⁶
GENERAL AVIATION ²	36.7 x 10 ⁻⁶	1.5	.23/.15	.6	7.6 x 10 ⁻⁶ /5.0 x 10 ⁻⁶
USAF FIGHTERS, COMPOSITE (F111, F101, F4)	62 x 10 ⁻⁶	2	.28/.17	.6	21 x 10 ⁻⁶ /13 x 10 ⁻⁶
USAF CARGO ³	2 x 10 ⁻⁶	4	.28/.17	.6	1.3 x 10 ⁻⁶ /0.8 x 10 ⁻⁶

RECOVERY AND TAKEOFF ACCIDENTS

1973

Table 3.1

1. Ref. 12, pp 38, 39
2. Ref. 11, pp 31, 32
3. Ref. 13

Then for a 24-hour mission,

Probability of Mission Abort = 0.032

not considering aborts which occur on the ground prior to takeoff during preflight.

3.6 Civil Airspace Hazards

Loss-rate probabilities fail to quantify potential hazards resulting from routine RPV operations within civil airspace. These hazards result from an RPV collision with either the ground or manned aircraft in flight.

3.6.1 Ground Hazards

Whenever a FCS failure causes an RPV to lose control and crash into a populated area, loss of life and destruction of property, of course, are possible. A probability of hazard could be calculated by multiplying the appropriate vehicle loss rate probability by a factor representing the relative population density beneath the vehicle. For example, it is estimated that 3%¹ of the area of the U.S. is occupied by people and property. Thus a factor of .03 would be appropriate for high-altitude (cruise) operations. On the other hand, a value of 0.5 might be appropriate for takeoff and recovery operations within a 5-mile radius of the typical airport because of the high population levels now encountered surrounding civil airfields.

Unfortunately, accepted hazard probability values have not been found against which to measure RPV performance.

To minimize the ground hazard, it is desirable to remove an RPV from populated areas in the event of total data-link loss. An RPV with which all communication has been lost cannot safely be brought home under automatic guidance. Instead, an alternate recovery procedure must be initiated over an unpopulated area and away from air corridors. Such an alternate-recovery program, tailored to the particular operating area, must be stored on board the vehicle. In the case of the YQM-98A evaluation in the Miami area, the alternate recovery program required flying an easterly heading out to sea for 60 minutes. If communication had not been restored at that time, the vehicle would be destroyed.

3.6.2 Midairs

It is, perhaps, even more difficult to quantify the midair collision hazard of a RPV operating among manned civil and military aircraft. At a minimum, equal means must be provided the RPV and its RO as are provided a manned aircraft and its pilot for avoiding midairs.

¹. Ref. 14

The problem with an RPV, of course, is its inability to see other aircraft. When operating IFR under IFR conditions, ATC can provide an RPV as good separation as provided manned IFR aircraft. When operating IFR in VFR conditions, however, the see-and-be-seen requirement applies. In the YQM-98A evaluation a U-2 chase plane provided the "eyes" for the RPV. This, clearly, would not be an acceptable procedure¹ when Compass Cope is operational. Thus a remoted visual capability with high resolution should be required.

3.7 Summary of Reliability Requirements

The suggested reliability requirements derived above are summarized in table 3.2. The table gives both mature and immature values for all requirements. The immature values were scaled from the mature using the F-111 to F-4 loss-rate ratio given above in Section 3.3.

PROGRAM MATURITY LEVEL	SUGGESTED FCS PROBABILITY			
	VEHICLE LOSS			MISSION ABORT
	ENTIRE FLIGHT	RECOVERY	TAKEOFF	
NEW (F111)	0.0033	41×10^{-6}	25×10^{-6}	0.078
MATURE (F4)	0.0017	21×10^{-6}	13×10^{-6}	0.032

TABLE OF SUGGESTED
RELIABILITY REQUIREMENTS

TABLE 3.2

1. The military operations manual MIL-7610-4C now requires a chase plane for any drone operating in FAA airspace.

4.0 STUDY GROUND RULES AND ASSUMPTIONS

4.1 Flight Profile Model

The flight-profile model assumed for this study is shown in figure 4.1. It consists of an outbound and a return segment, each of which is about one hour long, and the cruise segment of 22 hours. There are also takeoff and recovery segments.

Using data from figure 5-18, reference 15, the time for the takeoff roll and the time to climb to h_c (50 feet) gives a takeoff exposure time of about 25 seconds (0.0069 hours).

The recovery is defined to start at an altitude h_{appr} , 1500 feet above the runway at the glideslope intercept. If a nominal approach speed of 100 knots and a 4-degree glideslope are assumed, the time from 1500 feet to 50 feet is calculated to be about 123 seconds (0.0341 hours). From 50 feet to touchdown the time is 4 seconds (0.0012 hours), neglecting flare time. Using the deceleration rate of 5 knots/second from reference 15, the time required to stop from a touchdown speed 100 knots is 20 seconds (0.0056 hr). These exposure times were used in the failure analysis of the various candidate configurations in Appendix B.

A 24-hour total mission time was assumed.

4.2 Operational Assumptions

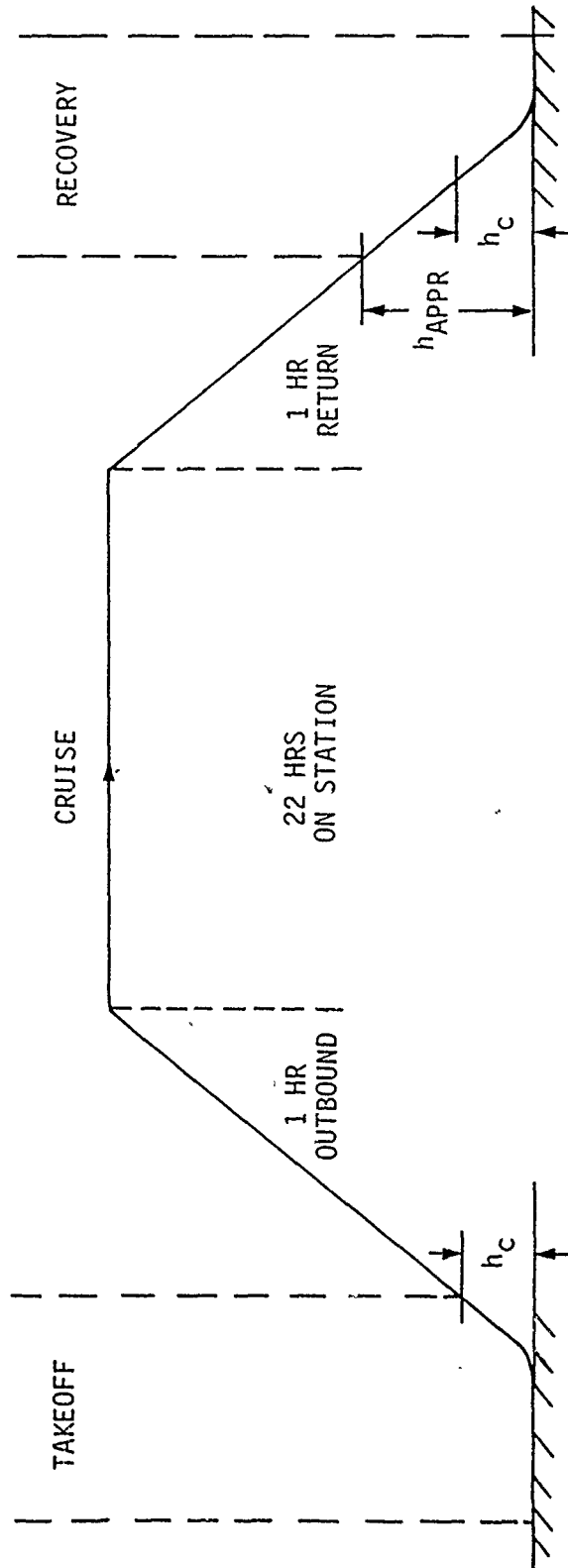
The following operational assumptions were used during this study:

1. A pre-programmed alternate recovery procedure will be initiated in cruise following total loss of data link.
2. An alternate recovery procedure will be initiated whenever the vehicle becomes unobservable due to a combination of data link and flight-critical sensor failures.
3. First failures on return leg do not count towards abort. Total failures on return leg do count towards vehicle loss.

4.3 Performance Assumptions

1. At a minimum, a basic attitude-hold, yaw SAS, and remote throttle control, which accepts up-link commands, is required for successful takeoff and recovery. This capability may be designated remote stick steering mode.
2. The vehicle can be remotely landed by the RO in remote stick steering mode under worst-case conditions with a 75% probability of success.

FIG. 4.1
FLIGHT-PROFILE MODEL



3. The remote operator cannot successfully assume control and revert to remote stick steering below a critical altitude, h_c . The value of h_c has been arbitrarily set at 50 feet¹ above the runway for both takeoff and recovery.
4. A fixed-pitch attitude is an acceptable backup control for go-around. Go-around is possible down to touchdown.
5. Aileron, elevator, rudder, and throttle control are flight critical during all segments of the flight profile.
6. Direct lift control is not a flight critical control.

4.4 Study Ground Rules

1. To simplify the analysis and the exposure time calculations, it was assumed that there is:
 - a) A 100% ground verification test prior to takeoff
 - b) A 100% self-test prior to the autoland phase of the recovery.
2. The MLS receiver and the radio altimeter are not powered above 10,000 feet. This improves the reliability and reduces the exposure time of these units.
3. Flights are not aborted after the first failure but allowed to continue for the full 24-hour mission duration. In actual practice, an RPV would immediately return to base after a first failure (flight control system, data link, or any first failure which precludes fail-operational status) to reduce its exposure time to a second failure. This groundrule simplifies the fault analysis, but yields pessimistic loss probabilities.
4. It is assumed that all undetected failures of either the hardware or the software are catastrophic. This is a very pessimistic assumption.
5. The sensors are not crossfed to the data links.
6. The dual sensors and control servos are 100% in-line monitored and have no unmonitored failures.
7. There are no losses in cruise due to the fault-free performance of the system. The remote operator can assume control if necessary.
8. Nuisance disconnects are included in the fault-free performance probability budget.

1. Since the completion of the redundancy study, critical altitude has been determined to be 200 feet for recovery and 50 feet for takeoff.

9. The level of fault-free performance achieved by the L-1011 automatic flight control system is assumed in the fault analysis for the control laws under consideration.
10. It is assumed that all failures are independent and non-conditional.

5.0 FAULT ANALYSIS APPROACH

The systems analyzed in this study are constructed from basic individual equipments. Each equipment is assumed to consist of a large number of components. Each component has a small probability of failing during an exposure time, T . The unit is considered to have failed when at least one component fails. The failures of all components are assumed to be Poisson distributed in time.

If A is the event that a unit fails during a flight or mission, then $P[A]$ is the probability of this event (a failure) occurring during a flight. Since the failures are assumed to occur randomly in time, their occurrence may be described by a Poisson distribution with a failure rate λ_F . Then

$$P[A] = 1 - e^{-\lambda_F T}$$

where T is the exposure time over which the failure may occur.

The series expansion of e^x is

$$e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

However, if $x \ll 1$, the series can be approximated by

$$e^x \cong 1 + x$$

Since $\lambda_F T$ is small for most cases, we can write the probability of failure equation as

$$P[F] \cong 1 - (1 - \lambda_F T)$$

$$P[F] \cong \lambda_F T$$

The following relationships from chapter 2 of reference 16 are also used in the failure analysis. Let A and B be two independent events in a sample space S . Then

$$P[A \text{ or } B] = P[A] + P[B]$$

$$P[A \text{ and } B] = P[A] \cdot P[B]$$

5.1 Fault Analysis Model

The top-level FCS fault analysis model used for this study is shown in figure 5.1. It shows the basic elements and the way they are interconnected. The data link is included as part of the system because vehicle losses can result from combinational failures of the flight-critical sensors and the data link. The need for an alternate recovery procedure (such as may occur due to a data link failure) is, for analysis purposes, included in the vehicle loss probabilities. The effects of the loss of the secondary flight controls are not considered in the fault analysis.

5.2 Vehicle Loss Fault Tree

A top-down fault tree concept was used for the fault analysis. The event of a vehicle loss is put at the top of the tree. The conditions and circumstances that contribute to the event are combined to feed into the resulting event. This type of diagram continues down until all reasonable fault conditions have been included. Only the top level fault trees are included in this section. The detailed, lower-level trees are included in Appendix B. The level of the fault trees in this section is general enough to discuss all the configurations analyzed in this study.

In the context of this report, the term vehicle loss due to an FCS failure will be a vehicle loss as a result of a failure of any of the elements shown in figure 5.1, except for the secondary flight control elements. Figure 5.2 shows how the probability of vehicle loss is divided into three component parts, takeoff, recovery and cruise. The probability for this can be written as:

$$P [\text{Vehicle Loss}] = P [\text{Takeoff Loss}] + P [\text{Recovery Loss}] + P [\text{Cruise Loss}]$$

The three component probabilities can further be divided down as shown in figures 5.3 through 5.5.

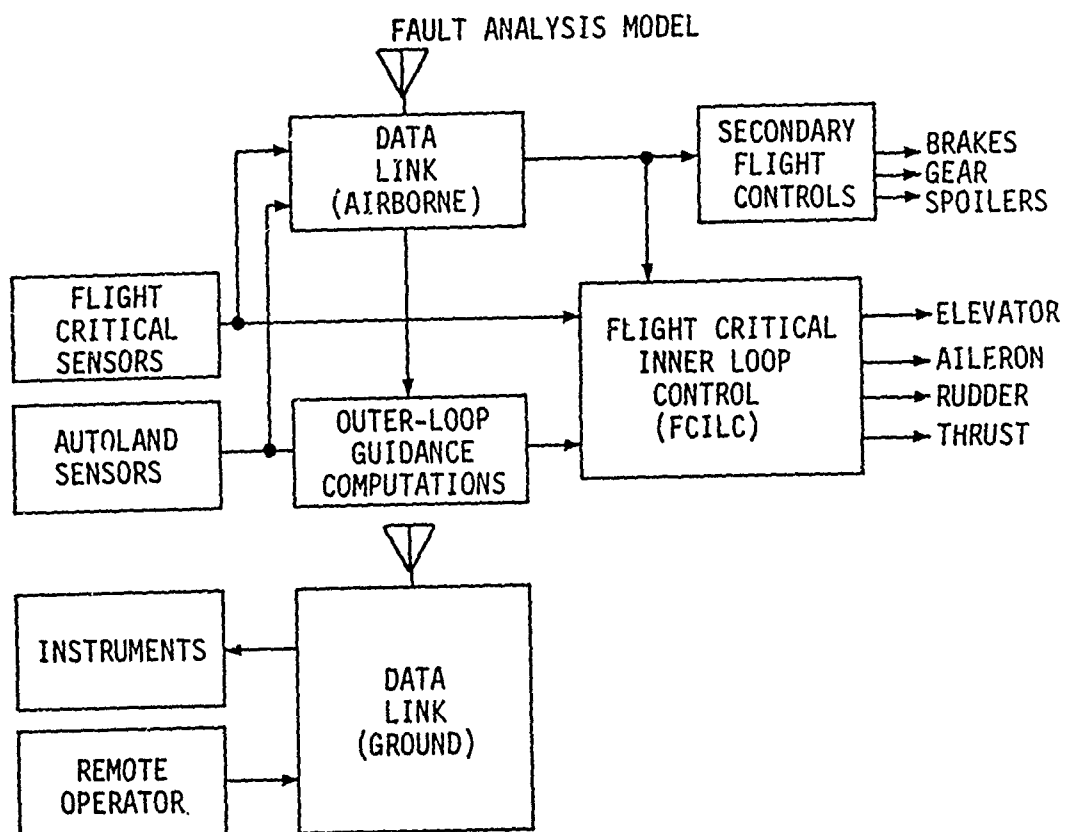
As an example of the analysis method, the recovery vehicle loss fault tree shown in figure 5.5 will be discussed here in detail.

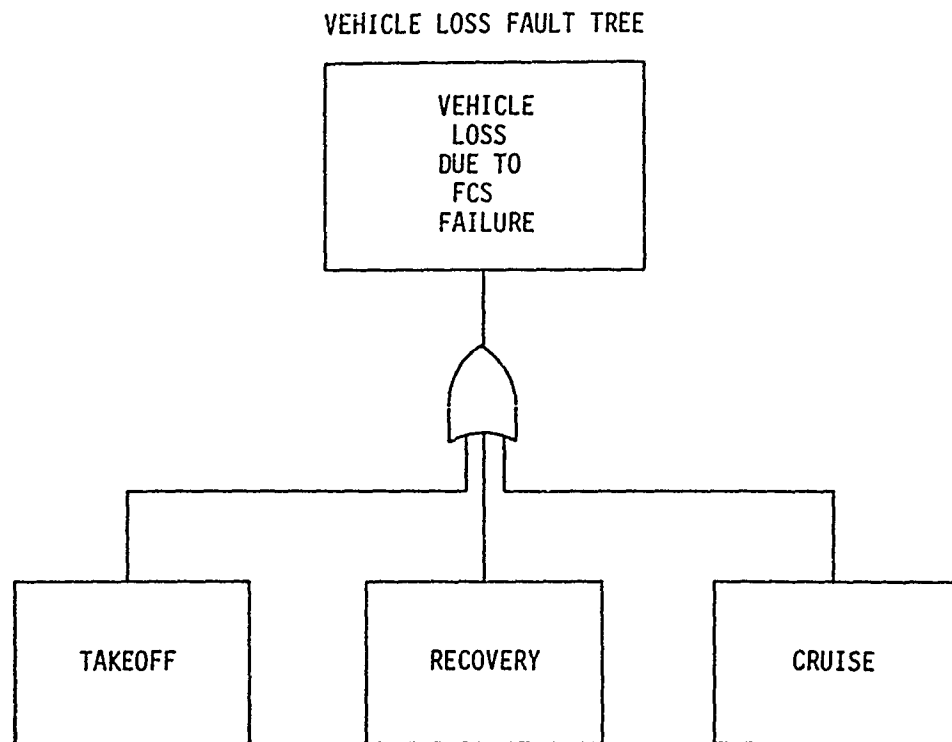
The top-level probability expression from the fault tree can be written as

$$P [\text{Recovery Loss}] = P [\text{FCILC Fails}] + P [\text{Data link Loss}] + P [\text{Loss above } h_c] + P [\text{Loss below } h_c]$$

The $P [\text{Loss above } h_c]$ conditions are those that can occur above h_c , but were not included in $P [\text{FCILC fails}]$ or $P [\text{Data Link fails}]$. The $P [\text{Loss above } h_c]$ is broken down into the conditions that guidance fails (requiring the remote operator to assume control) and, subsequently, the remote operator fails to make a safe landing.

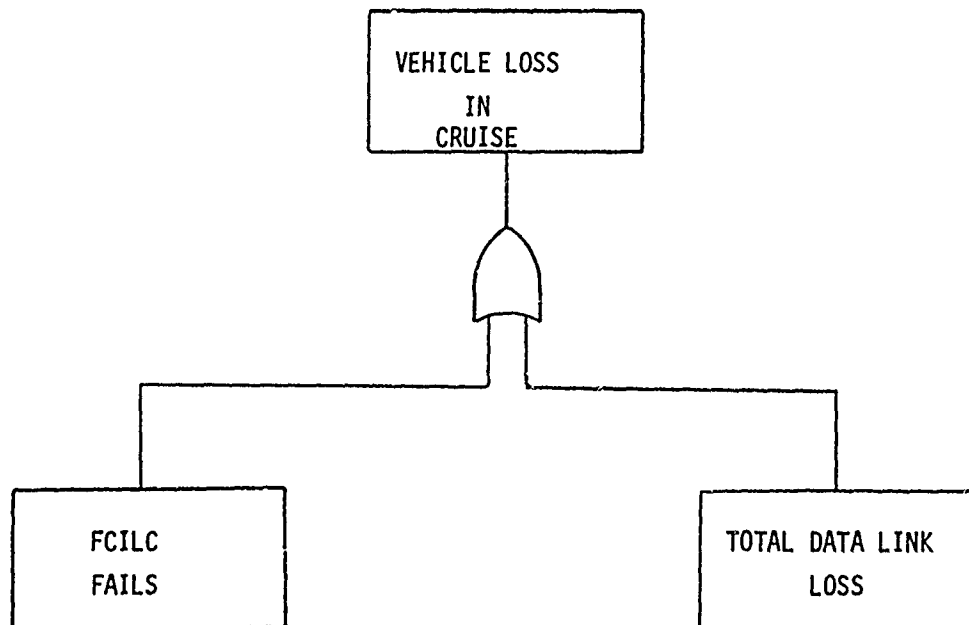
FIG. 5.1





VEHICLE LOSS FAULT TREE

FIG. 5.2

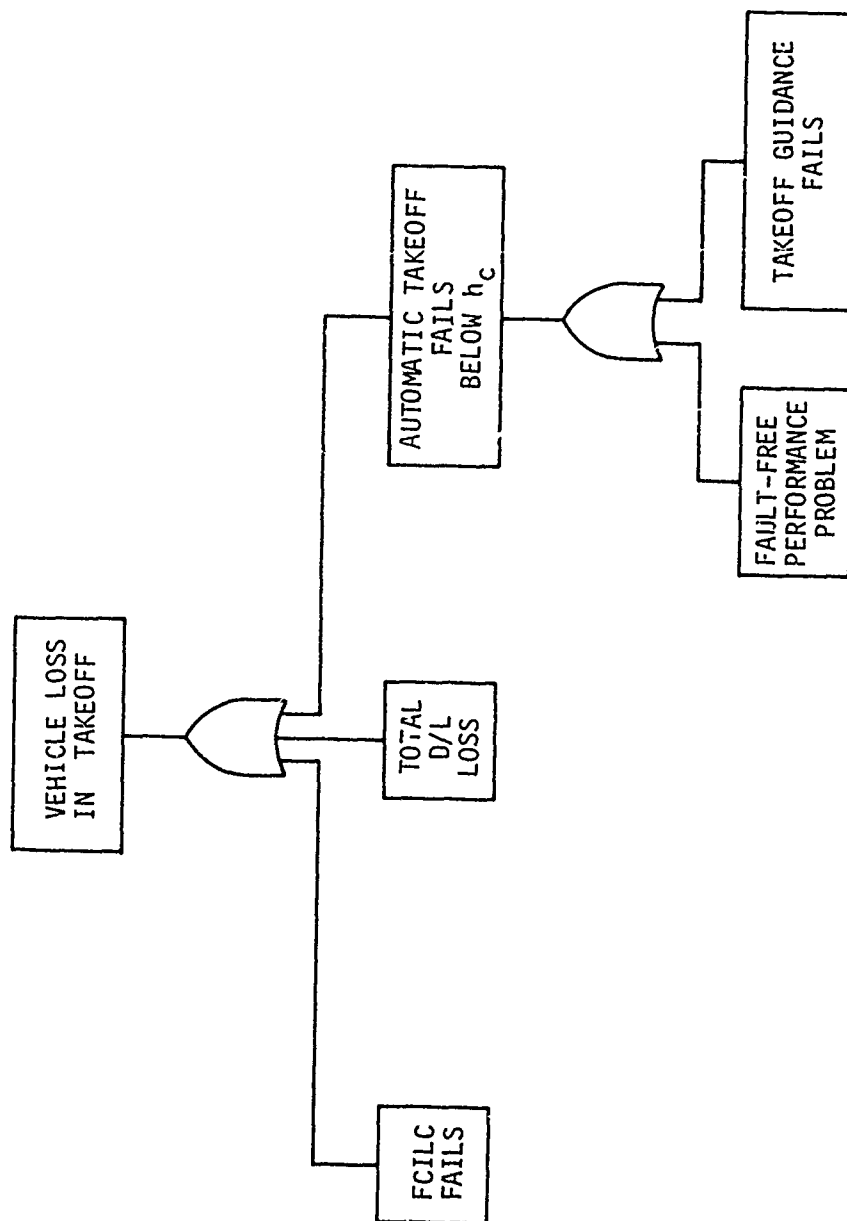


VEHICLE LOSS IN CRUISE

FIG. 5.3

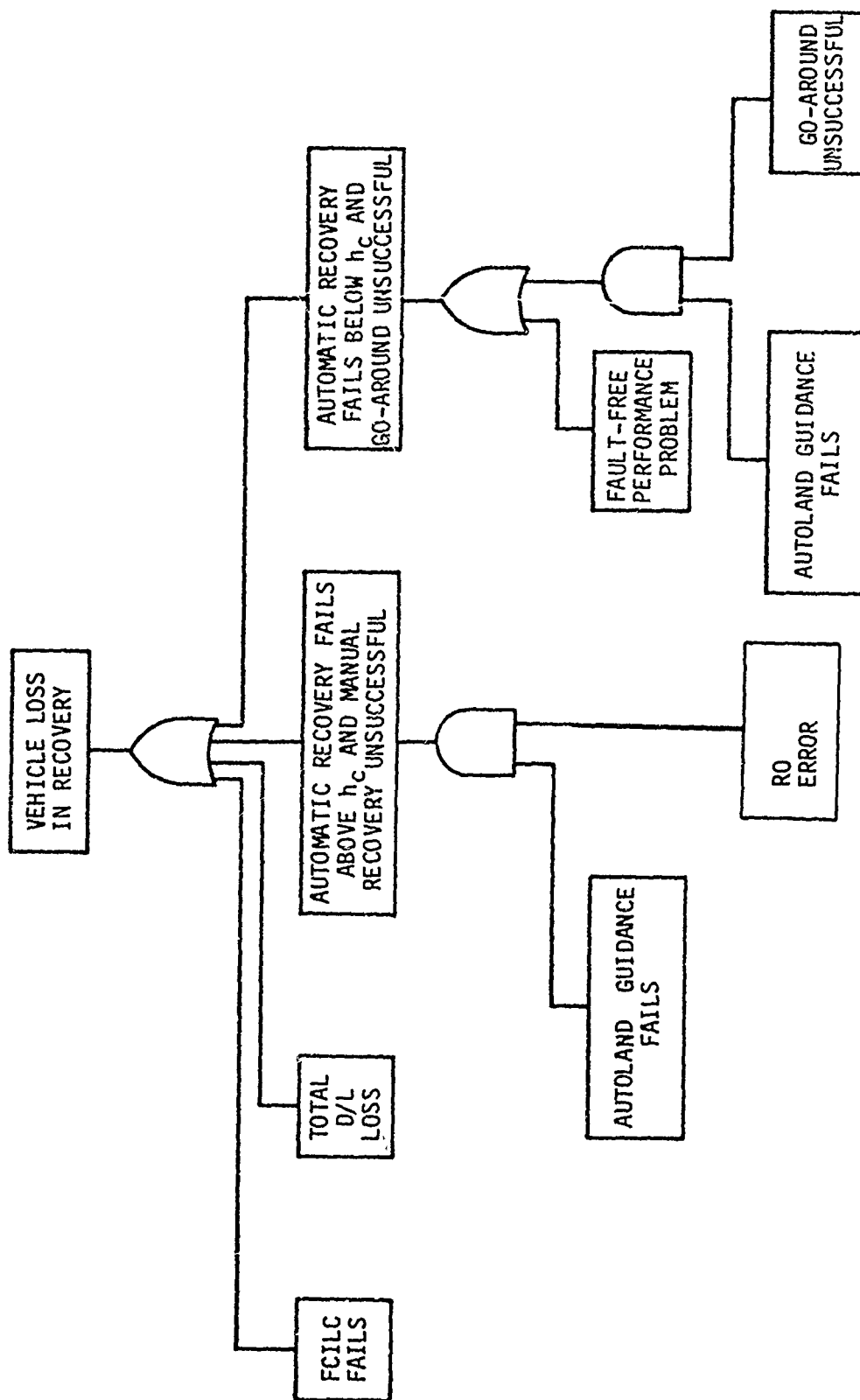
VEHICLE LOSS IN
TAKEOFF DUE TO
FCS FAILURE

FIG. 5.4



VEHICLE LOSS IN RECOVERY DUE TO FCS FAILURE

FIG. 5.5



$$P \left[\text{Loss above } h_c \right] = P \left[\text{Guidance Fails} \right] \cdot P \left[\text{R.O. Error} \right]$$

Below h_c it was assumed that the remote operator could not safely assume control of the vehicle. However, for detected failures a go-around could be initiated at any time all the way to touchdown. Below h_c the fault-free performance of the autoland computations can contribute to a vehicle loss and is added into the total. The go-around computations also have a fault-free performance probability which is included, but not shown, (refer to Appendix B) in the ability to perform a successful go-around. The vehicle loss below h_c can be written as:

$$P \left[\text{Loss Below } h_c \right] = P \left[\text{Fault Free Performance} \right] + P \left[\text{Guidance Fails} \right] \cdot P \left[\text{G/A Unsuccessful} \right]$$

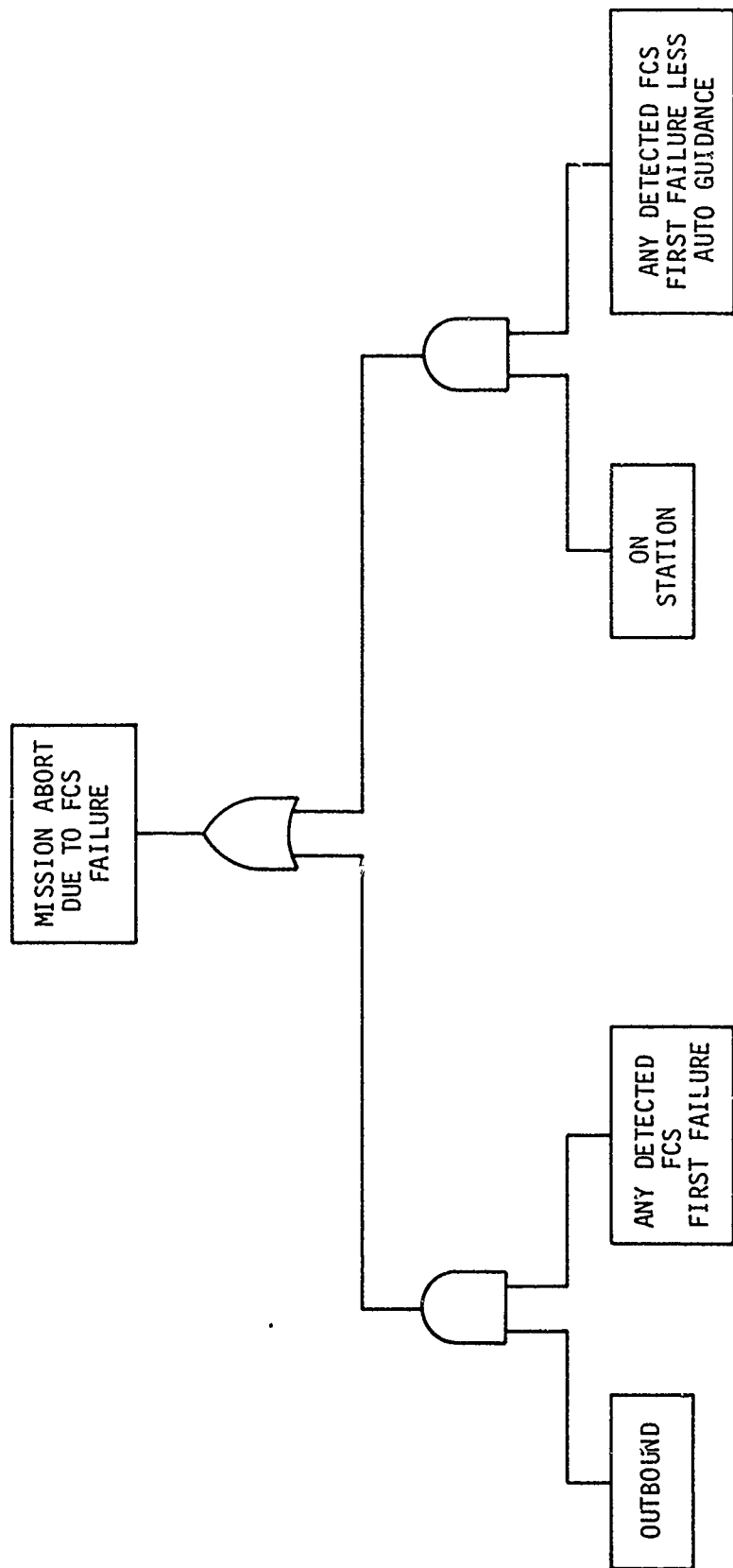
The next level down in the fault trees is configuration-dependent and can be found in Appendix B.

5.3 Mission Abort Fault Tree

The mission abort fault tree is generated in a manner similar to the vehicle loss fault trees. As stated in section 4, the failure of any unit in the system is a reason for a mission abort. Figure 5.6 shows the fault tree for the mission-abort probability.

MISSION ABORT

FIG. 5.6



6.0 ANALOG VS. DIGITAL HARDWARE

6.1 Flight Computer

Of the three basic classes of equipment comprising the Compass Cope flight control system, namely, sensors, flight computers, and servos, the flight computers are the most likely candidates for digital technology. State-of-the-art sensors and servos, with the exception of air-data computers, are typically analog devices.

Considering both types of technology, a strong case can be made for digital flight computers on the basis of size/weight/cost reductions and inherent advantages of digital technology.

6.1.1 System Size/Weight/Cost Reductions

The application of digital technology to the complex RPV problem will result in fewer LRU's, reduced weight, lower system costs. This is illustrated in Table 6-1 which shows a comparison of analog versus digital implementation for an air transport automatic landing flight control system. Additionally, a 33% improvement in system MTBF has been estimated for a transport digital FCS over an analog FCS.

The Collins systems¹ compared in Table 6.1 do not include as large a total mission computation requirement as for the Compass Cope flight computer which includes functions such as navigation, total system status monitoring, and data link interface processing. Adding these functions would represent an estimated 30% increase in the total problem complexity and would reinforce the advantages of digital implementation.

With the rapid movement in the avionics industry towards digital implementation, all application areas are reaping the benefits of large volume production of digital components. Digital component costs are expected to continue to decline at a rate faster than analog. This is illustrated in Figure 6-1 which compares equivalent function costs for an air transport autoland flight control system.

6.1.2 Inherent Advantages of Digital Technology

The inherent advantages digital technology offers over those of analog include:

Greater Computational Capability

Application of digital computer technology provides greater computational capability than can practically be obtained in an analog system. Combined with digital intersystem communication, each input sensor can be processed and voted, reducing the effects of sensor tolerances, and allowing maximum fault survivability with failed sensors. An analog input voting configuration would consume five 3/4 ATR cards and sixty pins per computer. The equivalent function in a digital implementation would represent approximately six pins, one card, and a software-implemented voter algorithm. Without digital implementation, sensor input voting would not be practical.

1. The FCS-110 analog system was certified in the Lockheed L-1011 transport. The FCS-111X digital system is currently being evaluated as part of a Boeing 7X7 study program.

	FCS-110 ANALOG SYSTEM	FCS-111X DIGITAL SYSTEM	IMPROVEMENT
SYSTEM VOLUME	7 X 3/4 ATR	3 X 1 ATR	57%
SYSTEM WEIGHT	140 lbs	120 lbs	14%
LIFE-CYCLE COST (10-year)	N	0.84 N	16%
LRU TYPES	4	1	75%
CARD TYPES	75 3/4 ATR	23 1/2 ATR	70%
SYSTEM INTER- CONNECTIONS (LRU pins)	1756	804	54%

ANALOG VERSUS DIGITAL COMPARISON
FOR
REDUNDANT AIR TRANSPORT FLIGHT CONTROL SYSTEM
(WITH AUTOLAND)

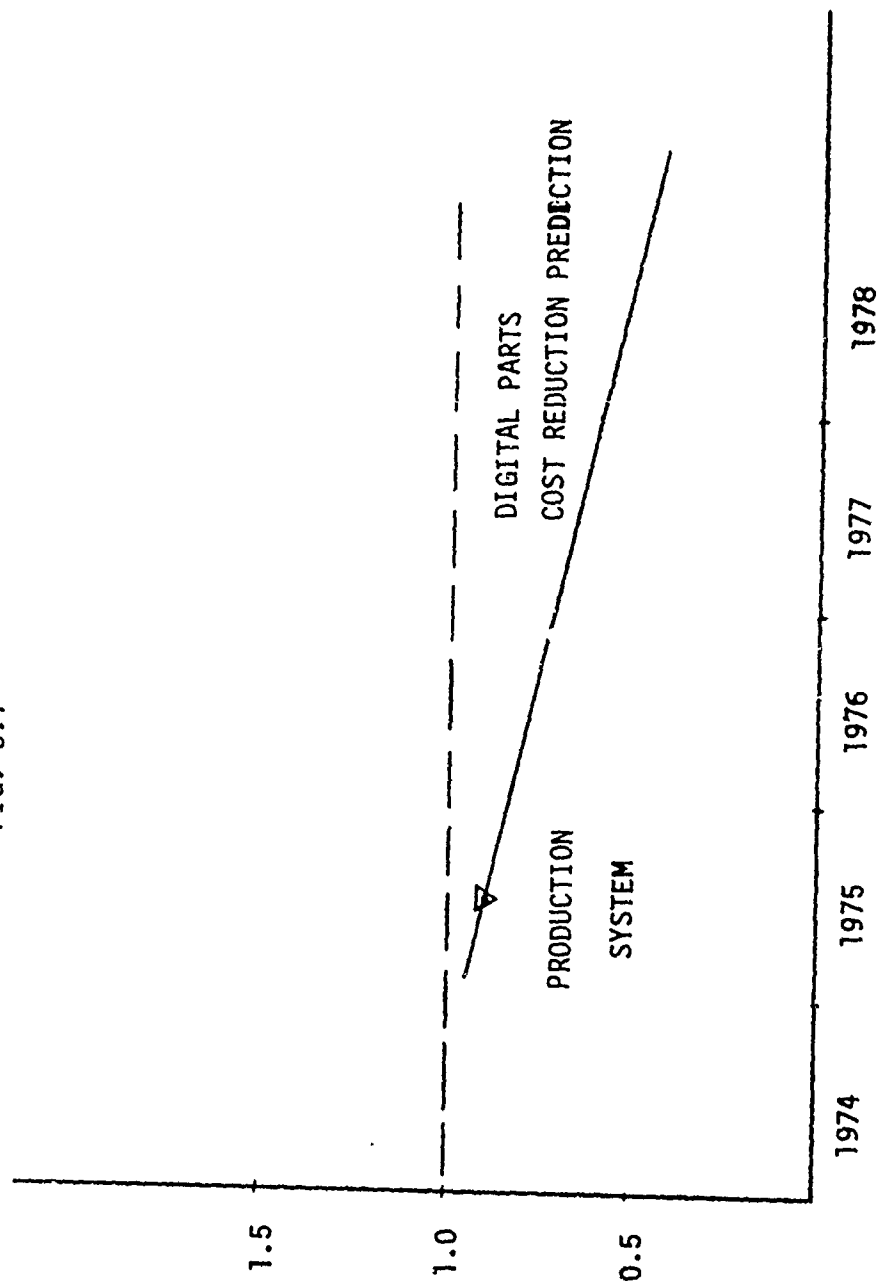
TABLE 6.1

DIGITAL COST COMPARISON

FAIL OPERATIVE FLIGHT CONTROL SYSTEM

DIGITAL/ANALOG
COST RATIO

FIG. 6.1



In addition, the computational capability of the digital computer allows use of sensor reasonableness tests which can provide a pseudo inline monitoring capability to improve fault survivability for sensor faults. This can be employed to provide increased survivability without adding additional sensors and computations which decrease the total system reliability.

Improved Self-Test Capability

A key advantage of digital implementation is the increased capability for system self-test and continuous monitoring relative to that available with analog implementation. Air transport applications of Built-in-Test (BIT) to a redundant automatic landing system have shown that coverage ranges from 75-90% with a ratio of BIT to total system hardware of 25-30%. A typical digital computer system self-test program would encompass 500-2000 words of memory and in a redundant system might represent a total of 2-3% of the total hardware. A high system BIT capability is essential for Compass Cope to accommodate the expected long mission durations. Dormant failures in the flight control system cannot be tolerated prior to takeoff or recovery phases.

System Flexibility

Digital systems offer the potential for a high degree of system flexibility with minimal hardware impact. This can reduce system design and development costs and provide a basis for easily accommodating system changes necessitated by mission profile and payload variations. Over the life of the vehicle, major subsystem components can be altered without complete redesign of the flight control system.

With the core digital processor, additional functions, such as total system status assessment and navigation guidance, can easily be accommodated via the addition of the necessary I/O and software modules.

Mission Adaptability

Digital systems offer the ability to accommodate changes in stored mission profiles without any hardware impact by simply modifying memory. This is a very important consideration for Compass Cope where mission profiles may vary from flight to flight.

7.0 DIGITAL PROCESSOR RELIABILITY AND MONITORING

7.1 General

The following sections consider the methods of digital processor self-monitoring, their effectiveness, and their costs. Both conventional and high levels of monitoring are evaluated to assess the cost of in-line monitoring. Failure probabilities are treated in the familiar way.

In studying processor monitoring, the following question arises: can processor hardware and software be separated for purposes of analysis? Certainly hardware and software failures can be measured separately, but the elusiveness of the unanticipated software bug makes it difficult to separate hardware and software monitoring. Conceptually, software and hardware can be assumed to be two statistically independent parallel sources of unit failures and can be combined in the usual manner to determine composite effect. But a perfect program, if such exists, contains no bugs and would display no failures. Therefore, there is theoretically no upper limit on software reliability. For this reason, the emphasis in this study is on processor hardware reliability, although in Section 7.4 a software reliability estimate is made to evaluate the effect of software bugs on FCS reliability. A digital processor reliability model which considers both hardware and software failure sources is shown in Fig. 7.1.

The following assumptions have been made:

Self-monitoring means monitoring of the basic processor: CPU, memory and a good part of the I/O.

This section is concerned with the effectiveness of digital flight computer self-monitoring and does not consider the system reliability implications.

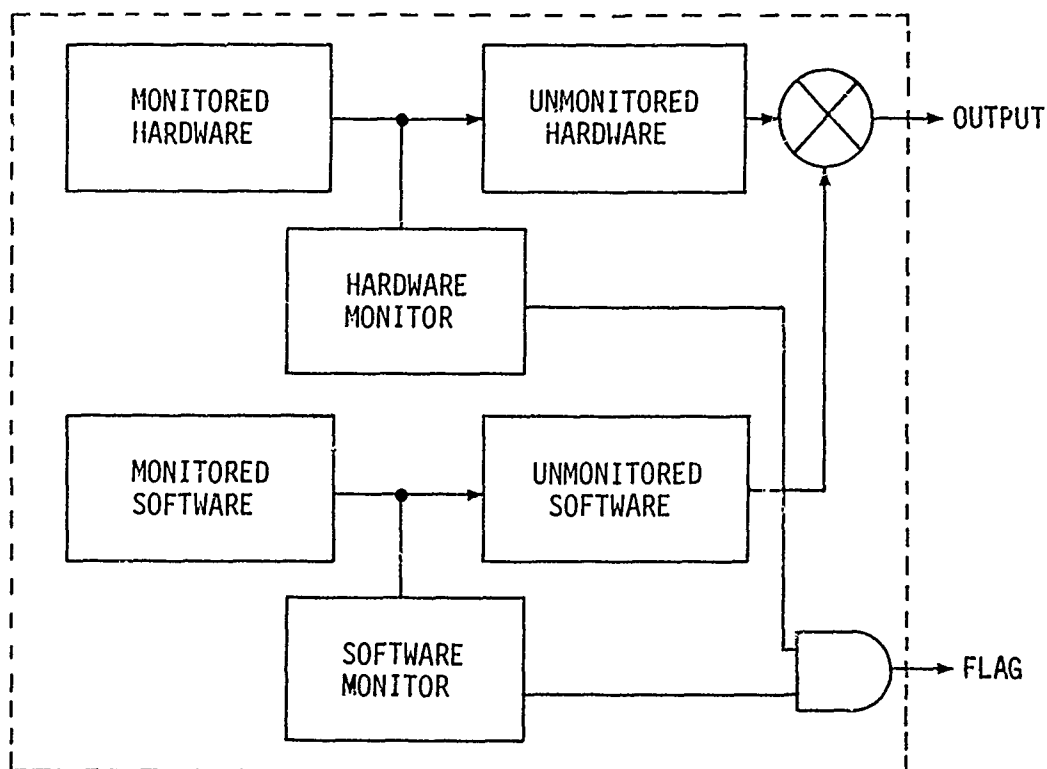
Test effectiveness percentages given in Section 7.3 are intuitive estimates only. Accurate values would have to be obtained by a thorough analysis on a fully-designed system (similar to Failure Mode Effect Analyses (FMEA) studies) to determine which failure modes are detected by each self-test feature.

It is assumed that the basic processor involved is the Collins CAPS-4 processor; estimates presented in sections 7.3 and 7.4 are based on CAPS-4 technology.

A comment should be made here about the feasibility of 100% monitoring (claims of 100% monitoring have been made by others). Assurance of perfect monitoring is limited by the ability to anticipate all possible failure modes. Perhaps the logical way to estimate percent monitoring, therefore, is as a percent of total identified failure modes. (Under this definition, the possibility of 100% monitoring is not so remote.) A failure-mode list should be made the basis for monitoring system design. Undetected hardware failure modes will suggest new self-test features which, if incorporated, will in turn improve the monitoring level, thus converging on "100%" monitoring.

FIG. 7.1

DIGITAL PROCESSOR RELIABILITY MODEL



7.2 Self-Monitoring Methods

According to S. Osder¹ of Sperry Flight Systems, "Comparison monitoring and the resultant dual-dual or triplex fail-operative architectures are a legacy of analog technology which can be rejected as the basis of a digital AFCS design. ... a digital computer does not require a second digital computer to verify that it is computing properly." A close look at self-monitoring techniques, however, spoils the illusion that self-monitoring is "free," because these techniques require redundancy in the form of additional or customized hardware, or in the form of memory space occupied by self-check software, or some combination. The additional redundancy required for self-monitoring, however, costs much less than duplicating the entire processor.

The following is a list of techniques that have been previously used for avionics processor self-monitoring (at Collins Radio, Sperry, Honeywell, Delco and others) together with brief descriptions of each. The techniques are classified according to the primary portion(s) of the processor or system that they are designed to monitor.

7.2.1 CPU Tests

1. CPU instruction set test: A self test program exercises the instruction set of the machine (usually the entire op-code set) and checks for valid results. Sometimes (e.g., MAGIC III) operands are generated by a pseudo-random number generator in order to produce a thorough test.
2. In-flight diagnostic (Collins 8564): A timed interrupt triggers a fixed sequence of test routines. The hardware keeps a tally of the number of control states executed. The diagnostic must complete in a specified time period and the exact number of total control states must be correct.
3. Redundant arithmetic hardware (MAGIC): Hardware is included in data path/arithmetic logic for fault detection and isolation. An example is the use of residue arithmetic for an adder check.
4. Transfer Bus Error Interrupt (CAPS-4): An interrupt is generated whenever the transfer bus is hung-up due to lack of response from memory, I/O, etc. This could also be a result of an improper address being presented on the bus by the CPU.
5. Miscellaneous CPU hardware checks (CAPS-4): Overflow detection, stack limit monitoring, and checks of the goto and nonlocal instructions are designed into the CAPS hardware.

7.2.2 MEMORY Tests

6. SUM checks on memory: A routine totals the actual contents (word-by-word) of a block of memory and compares the resulting sum to the pre-determined correct value.
7. Redundant Computations: Computations are duplicated either in separate memory modules (e.g., two 8K core memory modules rather than one 16K module) or in different parts of memory. Results are compared for validity.

8. Parity check: Memory instruction and data words are parity checked to provide detection of failures in memory and associated electronics.

7.2.3 I/O Checks

9. Constant voltage checks: Fixed power supply voltages are converted and compared to reference constants.
10. A/D - D/A loop closing: Analog outputs are fed back to inputs and, after conversion, are checked by the CPU against the original digital values.
11. Transmit and check test words: Fixed test words are periodically transmitted to test digital communication paths.
12. I/O Cross-compares: Converted inputs are digitally transmitted to other processor(s) and comparison-tested for validity.
13. Servo model: Servo actuator responses are monitored, modelled, and compared to check correct operation.

7.2.4 Software and System checks

(The following five items represent five variations of an approach which takes the electronic pulse of a processor. This single concept is quite effective in detecting a wide variety of processor hardware and software failures, especially those causing the machine to "lock up" or to fail to proceed through its instruction stream in the proper fashion.)

14. MFM "Machine Failure Monitor" (Collins C8561): An external machine failure monitor must be reset periodically to avoid an MFM alarm by executing a Reset MFM instruction.
15. Watch Dog Timer (Honeywell): A monostable flip-flop which, if not updated periodically, times out and disengages servos through hardware logic.
16. "Computer Operation (COP) Circuit" (Delco): The program must issue two alternating reset signals at a regular rate. An alarm occurs if the signals are late or fail to alternate.
17. Dynamic Computation Monitor (Honeywell): The processor generates a triangular waveshape by periodically alternating polarity of input to an analog integrator. If the processor fails to do so, a limit detector generates disengagement.
18. Hardware Pattern Monitor (Sperry): The processor must generate a correct dynamic output pattern of bits to an external hardware monitor.
19. DONE Check (CAPS): A "done" flag is reset at the beginning of a computation interval (as controlled by a timed interrupt) and is set at the end of the computation. When the next timer interrupt occurs, the done flag is tested to verify that it is set indicating that the computation completed properly before being interrupted for a new cycle. This feature is quite similar to the MFM monitor, but requires no external hardware.

20. Software Task - Done Monitor (Sperry): Similar to the DONE check, but with more resolution. A separate flag is set for each task in a sequence of tasks comprising the main computation. This could be combined with the CPU test (1) and hardware pattern monitor (18) to provide a detailed in-flight verification of the processor's CPU logic and timing, together with a certain amount of software monitoring.
21. Power Failure Interrupt: Interrupts the CPU in the event of a power interruption.
22. Reasonableness checks and validity checks on sensor data: Provides a performance monitoring mechanism for sensors and also verifies a portion of the I/O operation.

7.3 Processor Hardware Reliability

This section attempts to assess the level of monitoring obtainable with the recommended self-monitoring techniques and to estimate the resulting cost impact (both recurring and non-recurring). A CAPS-4 processor was used for analysis purposes.

In order to limit the problem, two levels of monitoring were considered:

1. A conventional-level capability consisting of those features available at little or no additional cost in a CAPS-4 processor. This was defined to include a minimal CPU test (method 1 of section 7.2.1) together with the self-checks numbered 4, 5, 6, 9, 10 and 19.
2. A high-level capability consisting of the conventional plus a software Task Monitor (method 20) incorporating a thorough CPU test and replacing the DONE check, plus a Hardware Pattern Monitor designed in such a way that the pattern of bits is dependent on the results of the CPU self-checks, plus self-check features numbered 7, 11 and 13 above.

Table 7.1 presents estimates of monitoring effectiveness of the various techniques, itemized into CAPS-4 sub-system elements. Table 7.2 shows estimates of incremental recurring cost (in dollars) and non-recurring cost (in man-months) of each feature. Software recurring cost estimates were made by prorating memory cost according to the size of the programs involved.

Table 7.3 uses failure-rate data together with estimates from Table 7.1 to derive MTBF values for the conventionally-monitored and highly-monitored configurations.

Figure 7.2 presents a gross picture of how monitoring level varies with additional costs of implementing/incorporating the monitoring features. Here it was assumed that the two extremes of the curves were the conventionally-monitored and well-monitored configurations discussed above. Progressing from conventionally-monitored to highly-monitored levels, those features were added first which produced the most failure detection for the least increased cost; thus causing the curves to steepen from left to right.

SELF-CHECKING TECHNIQUE EFFECTIVENESS

CAPS-4 PROCESSOR

% CHECKED

TECHNIQUE	CORE MEMORY	IN- PUT	OUT- PUT	MICRO CONTROL	DATA PATH	TRANSFER INTERRUPT
Memory Sum Checks	70					75
A/D, D/A loop close & constant-voltage checks		85	85			
Minimum instruction tests + done check				(50)	(50)	
Xfr buss timeout, stacklimit, overflow, go-to, etc.	5	5	5			5
Software task monitor	x } 1			x } 95	x } 95	x } 10
Hardware pattern monitor*	x }		2	x }	x }	x }
Redundant Memory *	20					5
Check words	.5	2	3			

TABLE 7.1

*Not needed in microprocessor monitor approach.

TABLE 7.2
INCREMENTAL COST OF SELF-CHECKING TECHNIQUES

CAPS-4 PROCESSOR

TECHNIQUE	ESTIMATED RECURRING COST PER SELF-CHECKING TECHNIQUE	ESTIMATED NON-RECURRING EFFORT (MAN MONTHS)	
Memory Sum Checks	\$26 (30 words)	0.5	↑ AVAILABLE AT LITTLE OR NO COST
A/D, D/A loop close & constant-voltage checks	\$50	0.2	
Minimum instruction tests + done check	\$170 (200 words)	1.5	
Xfr buss timeout, stacklimit, overflow, go-to, etc.	Currently implemented	- -	↓ AVAILABLE AT ADDITIONAL COST
Software task monitor	\$350 (400 words)	4	
Hardware pattern monitor*	\$150	1.3	
Redundant Memory*	\$350 (400 words) 1000 Mem cost 400 Interface \$1750	2	
Check words	\$85 (100 words)	0.9	↓

*Not needed in microprocessor monitor approach.

TABLE 7.3

ESTIMATING LEVEL OF DFC SELF MONITORING

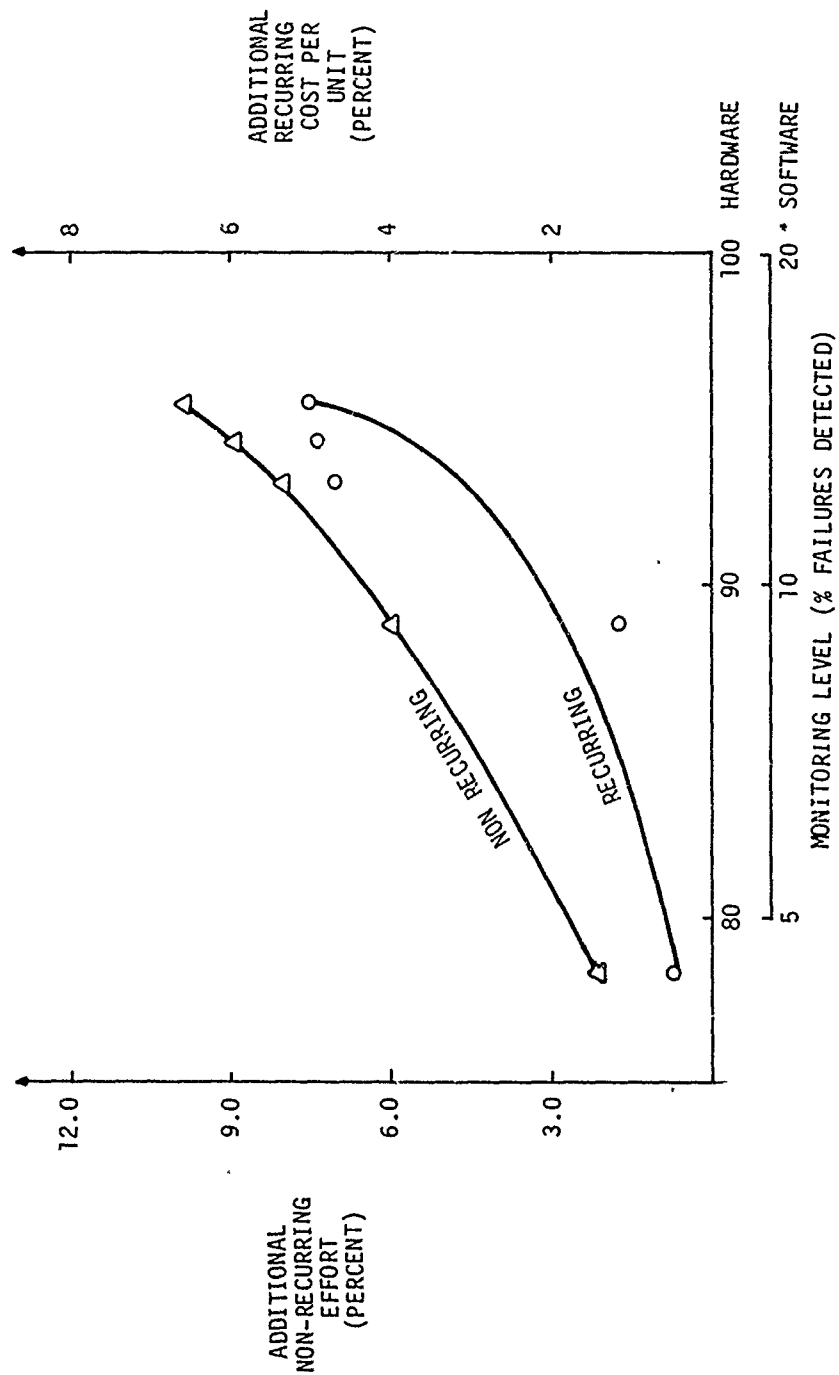
ELEMENT	FAILURE RATE	CONVENTIONAL LEVEL		HIGH LEVEL	
		% CHECKED	UNDETECTED FAILURE RATE	% CHECKED	UNDETECTED FAILURE RATE
SEM-9 memory	5×10^{-5}	75	1.25×10^{-5}	97	1.5×10^{-6}
I/O	1.34×10^{-4}	85	2.01×10^{-5}	95	6.7×10^{-6}
Microcontrol	2.1×10^{-5}	50	1.06×10^{-5}	95	1.05×10^{-6}
CPU	2.9×10^{-5}	50	1.47×10^{-5}	95	1.45×10^{-6}
Transfer Buss Interface & Interrupt Logic	1.45×10^{-5}	80	2.9×10^{-6}	95	7.25×10^{-7}
Totals	2.485×10^{-4}		6.08×10^{-5}		1.143×10^{-5}

FAILURE RATE SUMMARY

Basic DFC = 2.485×10^{-4} Unmonitored Parts, Conventionally - Monitored DFC = 6.08×10^{-5} (75%)Unmonitored Parts, Highly - Monitored DFC = 1.143×10^{-5} (95%)

FIG. 7.2

INCREMENTAL
COST OF
IN-LINE MONITORING



REFERENCE PROCESSOR: COLLINS MILITARY CAPS-4 PROCESSOR, 16K MEMORY

Again it should be emphasized that much of the above data concerning monitoring effectiveness was obtained intuitively, therefore evaluation of a finalized system should involve a thorough inspection of specific failure-mode information.

7.4 Software Reliability

Although software reliability is of growing concern to those involved in software engineering, the nature and frequency of software "failures" are not nearly as well understood as the corresponding hardware discipline. According to a recent study by TRW, software errors in design outnumber coding errors almost 2:1. Whereas most coding errors are found before acceptance testing, the vast majority of design errors are discovered during or after acceptance testing. It would seem that bugs remaining in mature programs would be due either to incomplete testing of the program or to unanticipated "stress" (unexpected state of the environment or the program), or to insufficient specification, understanding or communication of the program's required function.

Consider the problem of software fault monitoring and placing a quantitative measure on the effectiveness of such monitoring. Although some of the methods discussed in 7.2 above will certainly detect some software failures, it is very difficult to identify failure modes for software a priori. On the other hand, since there is no theoretical upper limit on the attainable software reliability (i.e., software devoid of bugs would have an infinite MTBF) the real concern should be with methods of producing more reliable software. To this end, software engineering researchers are recommending ideas such as the following:

- a) Use of high-level languages
- b) Structured programming techniques
- c) Top-down design
- d) Thorough debugging
- e) Limitations on module size
- f) Self-metric techniques

This discussion concludes with a brief note about what level of software reliability might initially be expected, using an extrapolation of data gathered by Miyamoto.¹

The referenced article describes measured MTBF for mature software (i.e., after acceptance testing) of 396.5 hours for a program roughly equivalent to 197,000 16-bit words.

Assuming failure rate to be directly proportional to program size allows the following speculation to be made concerning potential software reliability for 16K software:

$$\text{Software MTBF} = 396.5 \times 197\text{K}/16\text{K} = 4800 \text{ hours.}$$

The conventional and high monitoring levels might detect 5% and 20%, respectively, of the (strictly) software bugs. This would lead to the following estimated improvements in (undetected) failure rate, stated here only in terms of MTBUF:

Conventionally - monitored: software MTBUF = 5000 hours

Highly - monitored: software MTBUF = 6000 hours

7.5 Use of a Microprocessor as a Monitor

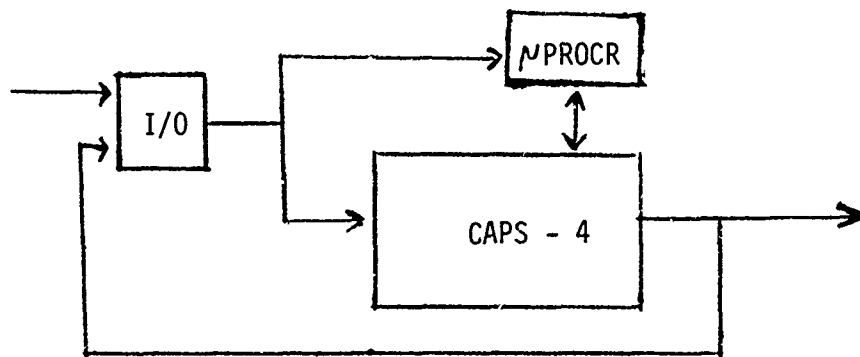
The highly-monitored configuration discussed in Section 7.3 included a hardware pattern monitor external to the processor. If the hardware pattern monitor is replaced by a microprocessor, fig. 7.3, several benefits result:

1. Probable cost saving with respect to hardware pattern monitor approach. Can tradeoff deletion of hardware pattern monitor and redundant memory computation (Tables 7.1 & 7.2) against additional cost of microprocessor.
2. Flexibility.
 - a. Could provide red-line monitor function
 - b. Could provide backup autopilot function
 - c. Could be a low-speed parallel computation channel with threshold checking
3. Could add tests as FMEA show necessary
4. Design could be streamlined if microprocessor shared a common high-level language with main processor

The microprocessor approach would provide some additional failure detection:

1. Some software checking and some I/O checking would be inherent
2. On-line verification of the monitor microprocessor by the main processor (i.e., cross-checking) would be possible.

The microprocessor approach looks appealing from both cost and capability viewpoints, and is recommended for the highly-monitored digital flight computer.



MICROPROCESSOR MONITOR

Fig. 7.3

8.0 SYSTEM CONFIGURATION CANDIDATES

It is shown via fault analysis in Appendix A that fail-operative sets of both sensors and servos are required to satisfy the FCS reliability requirements. It is then reasonable to generate system configurations by fixing the sensors and servo configurations and varying the inner-computation architecture.

There are two basic inner-computation architectures:

- A. A fixed fail-operative set of inner-loop control computations. Outer-loop guidance computation redundancy and monitoring are varied.
- B. Inner and outer-loop computations are combined in individual units. Redundancy and internal monitoring of the units are varied.

The second architecture, B, was selected for the redundancy study. A was discarded because it appeared on the surface to be more costly than B for the minimum configurations. An implementation of A would typically require three sets of analog computations, most likely packaged in three separate units. A minimum configuration of B, on the other hand, would require only two units. Time did not permit, unfortunately, in-depth analysis of A.

Five system configuration candidates were generated from the basic architecture of B. Starting with a fail-operational set of sensors and servos, flight computer redundancy, flight computer monitoring levels, auxiliary redline monitoring, and backup equipment were varied to generate the candidates, as shown in Fig. 8.1.

8.1 Configuration Components

A list of representative equipment from which the five candidates were constructed is shown in Table 8.1. State-of-the-art off-the-shelf equipment was selected when possible. New designs were selected only when off-the-shelf equipment was not adaptable.

8.1.1 Sensors

Triple vertical gyros, rate gyros, and accelerometers were used in the study to generate a fail-operative set, since these sensors are traditionally not highly-monitored. Pitch and roll rates are derived from the respective attitude signals within the flight computer(s). Yaw rate is supplied by triple rate gyros for the yaw SAS and runway alignment computations.

Dual air data computers, compass systems, MLS receivers, and radio altimeters were selected. The MLS receivers and radar altimeters are of autoland quality and are highly self-monitored. Dual CADC and compass systems were selected, even though they are less than perfectly monitored. Occasional undetected failures occurring in these sensors would not be catastrophic and could be accommodated by the RO.

Dual two-way narrow-band APQ-3 data link systems were selected for transmission of down-link telemetry and up-link autopilot steering commands.

8.1.2 Flight Computers

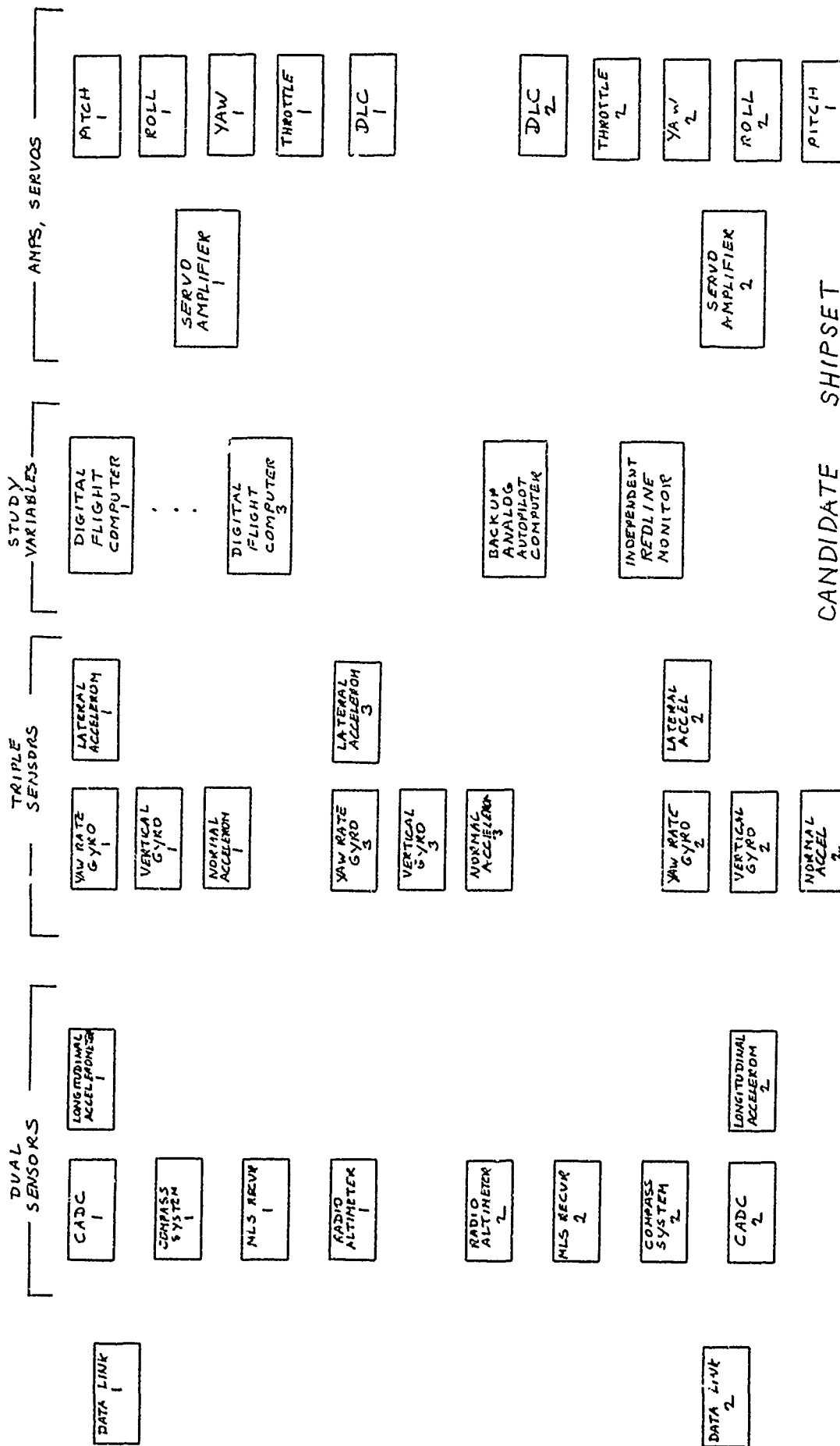
Two digital flight computers (DFC's) with different in-line monitoring levels were designated as study variables. A conventionally monitored computer was chosen to represent a standard off-the-shelf digital computer with a nominal in-line monitoring level of 75%. The other computer represents a computer designed to have the highest level of monitoring practically attainable. This level is estimated to be 95%. The term "flight computer" used in this study includes a digital processor and an aircraft systems coupler (ASC) all within one package. The desirability of a digital implementation for the flight computer has been shown in Section 6 above.

The flight computers provide both the autopilot and mission navigation computations for the Compass Cove FCS. The autopilot computations include a basic remote stick-steering control, various cruise modes, and automatic takeoff and recovery (autoland) guidance. The navigation computations include the various station-keeping and mission guidance, as well as preprogrammed alternate-recovery guidance for execution in the event of total loss of data link. Built-in test (BIT), sensor interfacing, and non-FCS redline monitoring are also important flight computer functions.

DESCRIPTION	TYPE NO.	MANUFACTURER	WT. (LBS.)	SIZE (IN)	MTBF (HRS)	UNIT PRICE	QTY/SHIP
Central Air Data Computer	A-1/ASK-6	Sperry	17	12 x 8 x 6	4,000	18,000	2
Compass System	ASN-89	Sperry	15	6 3/4 x 7 1/2 x 6.5	4,000	3,500	2
Vertical Gyro	9000-C	Leair	5	4 1/4 x 6	2,000	3,000	3
Radio Altimeter	860F-1	Collins Radio	20	15.5 x 7 1/2 x 5.2	1,600	7,000	2
Rate Gyro	60-17000	Sperry	0.57	1.5 dia x 3.75	22,000	1,500	3
M.S. Receiver	513J-1	Collins Radio	25	1/2 ATR Short	3,000	3,800	2
Accelerometer (Single Channel)	P/O 345C-1	Collins Radio	0.25	2 x 1.5 x 1.5	50,000 (Chnl)	700	8
Servo Amplifier (5-Channel)	SA-XX	Collins Radio	11	1/2 ATR Short	8,000 (Box)	8,400	2
Auto Pilot Servo	SVC-P0	Collins Radio	9.3	6 1/2 in ³	33,000	1,100	8
Auto Rotative Servo	3347-2	Collins Radio	3	3.95 x 4.6 x 6.0	20,000	1,100	2
Independent Red-Line Monitor	RLM-XX	Collins Radio	8	3/8 ATR Short	6,000	5,200	0/1
Backup Analog Flight Computer	10-XX	Collins Radio	19	1/2 ATR Short	1,000 (Box)	7,000	0/1
Narrow-Band Data Link System	APQ-3	Sperry Univac	14	NA	1,000	NA	2
Airborne Transponder (P/O APQ-3)	APR-26	Sperry Univac	24	720 in ³	4,900	16K	2
Digital Flight Computer	10-XX	Collins Radio	35	Full ATR	4000/16400*	60,000	1/2/3
CONVENTIONAL MONITOR (75%)	DFC-X2	Collins Radio	35	Full ATR	3800/8750*	65,000	1/2/3

*MTBF/MTDUF

REPRESENTATIVE COMPASS COPE FLIGHT CONTROL SYSTEM EQUIPMENT
TABLE 8.1



CANDIDATE SHIPSET

FIG. 8.1

8.1.2.1

The Conventionally-Monitored Computer

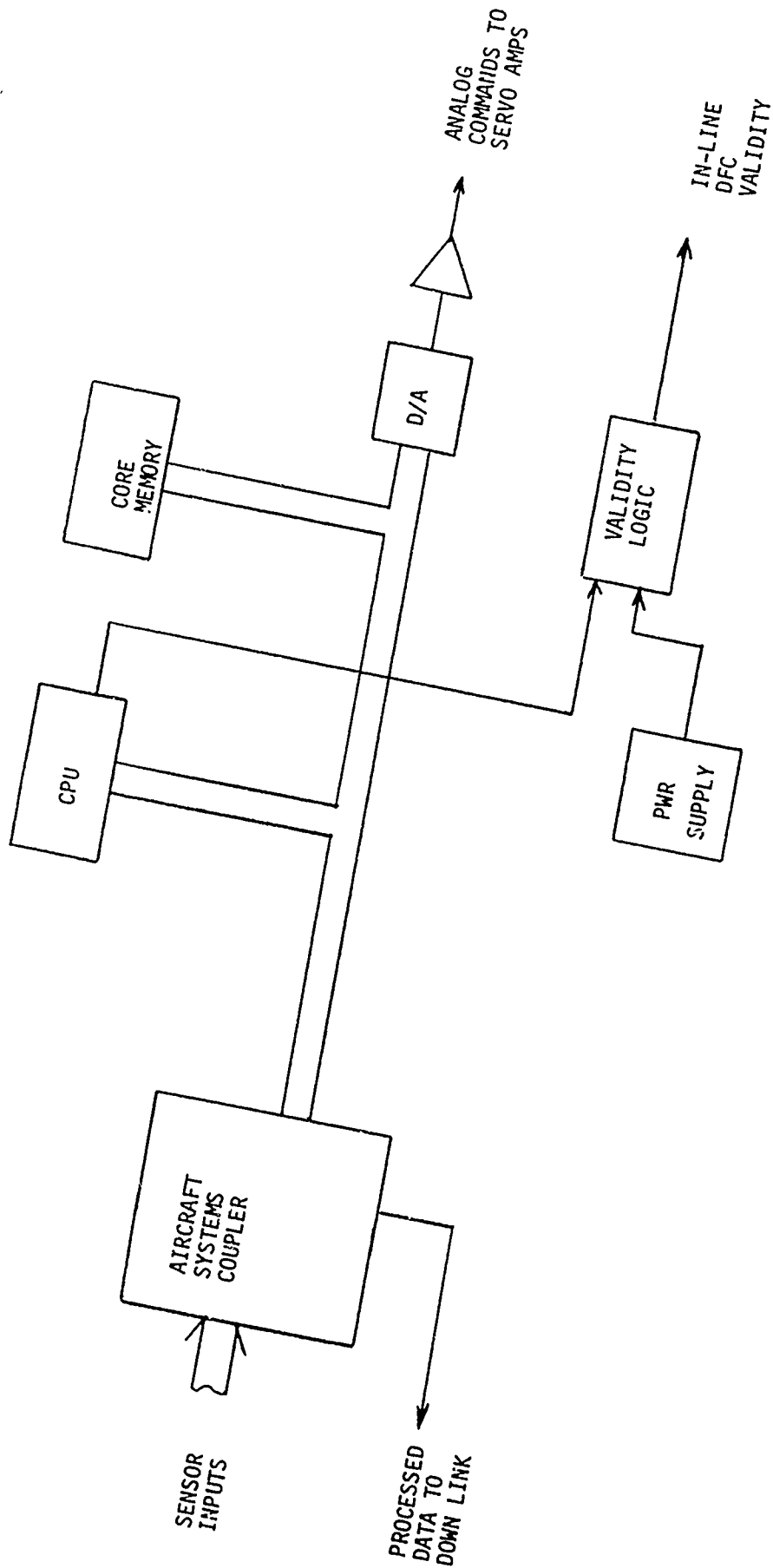
A monitoring level (percent of hardware failures detected) of 75% was felt to be attainable for "free" in a standard off-the-shelf digital flight computer. A Collins computer utilizing a CAPS-4 processor was selected for purposes of the redundancy study. The CAPS-4 is a stack-oriented microprogrammed machine with a speed of approximately 300 KOPS. A 16K-word core memory (16-bit word length) was assumed. The internal computer architecture is shown in Fig. 8.2.

8.1.2.2

The Highly-Monitored Computer

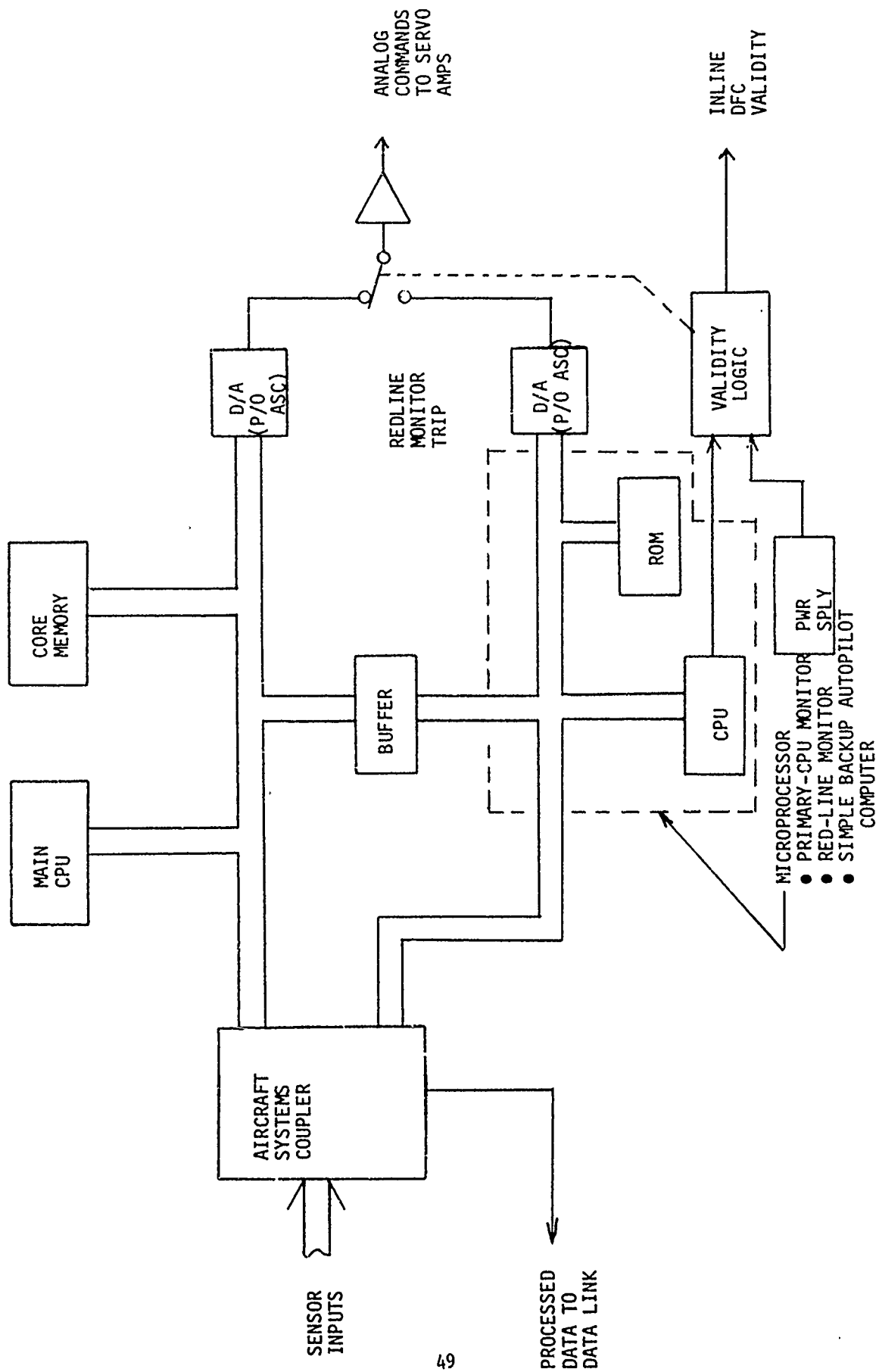
As discussed in Section 7.3, a relatively high monitoring level of 95% may be designed into a machine for amazingly small additional cost (5%). Furthermore, if a separate microprocessor is used to perform the required hardware pattern monitoring external to the main CPU, it becomes an independent processor available for other limited tasks. This requires, of course, a free-running I/O (ASC) to allow data flow independent of the main CPU and transfer bus.

Thus the configuration of 8.3 was chosen for the highly-monitored digital flight computer. Besides performing the required hardware pattern monitoring, the internal microprocessor performs red-line monitor and simple backup autopilot computer tasks. As discussed in Section 7.6, the red-line monitor can detect, in most cases, the effects of undetected processor hardware failures and software problems by monitoring aircraft performance. The red-line monitor switches the DFC output to a simple backup autopilot control. The backup control, because of limited microprocessor capacity, can be no more than basic attitude hold modified by up-link commands. The backup must also include a fixed-pitch go-around for problems occurring during auto recovery.



CONVENTIONALLY-MONITORED
DIGITAL FLIGHT COMPUTER

FIG. 8.2



HIGHLY-MONITORED DIGITAL FLIGHT COMPUTER
FIG. 8.3

8.1.3

Servos

Dual in-line monitored electro-mechanical servos were selected for each of the control surface and throttle servos in the candidate systems. Both torque summing and alternate-engage switching configurations were considered. The latter configuration is favored.

The servo fault analysis in Appendix A justifies the need for fail-operative servos. All control surfaces, except for DLC and throttle, were considered flight critical. A single in-line monitored servo would thus be adequate for DLC. Dual servos were selected for DLC, however, for reasons of symmetry. The servo complement is shown in the Candidate Shipset Diagram, Fig. 8.1. Time did not permit consideration of both hydraulic and electro-mechanical control-surface servos. Consequently, only the latter were considered.

As shown in Fig. 8.4, each dual servo channel is implemented with a single servo motor driven by dual servo amplifiers with dual tach feedbacks. Redundant cross-channel comparators provide the required in-line monitoring. Five control channels (pitch, roll, yaw, DLC, and throttle) are packaged in a single servo amplifier LRU.

Two side-to-side servo coupling techniques were considered. Fig. 8.4 shows an alternate-engage switching technique in which each servo is coupled directly to the control surface via its own engage clutch. Only one clutch may be engaged at a time, servo 1 being normally engaged and driving. When a failure occurs in servo 1, servo 2 engages and servo 1 disengages.

Fig. 8.5 shows another technique which employs torque summing via a mechanical differential. Since both servos are normally engaged, and driving, crosschannel equalization is required.

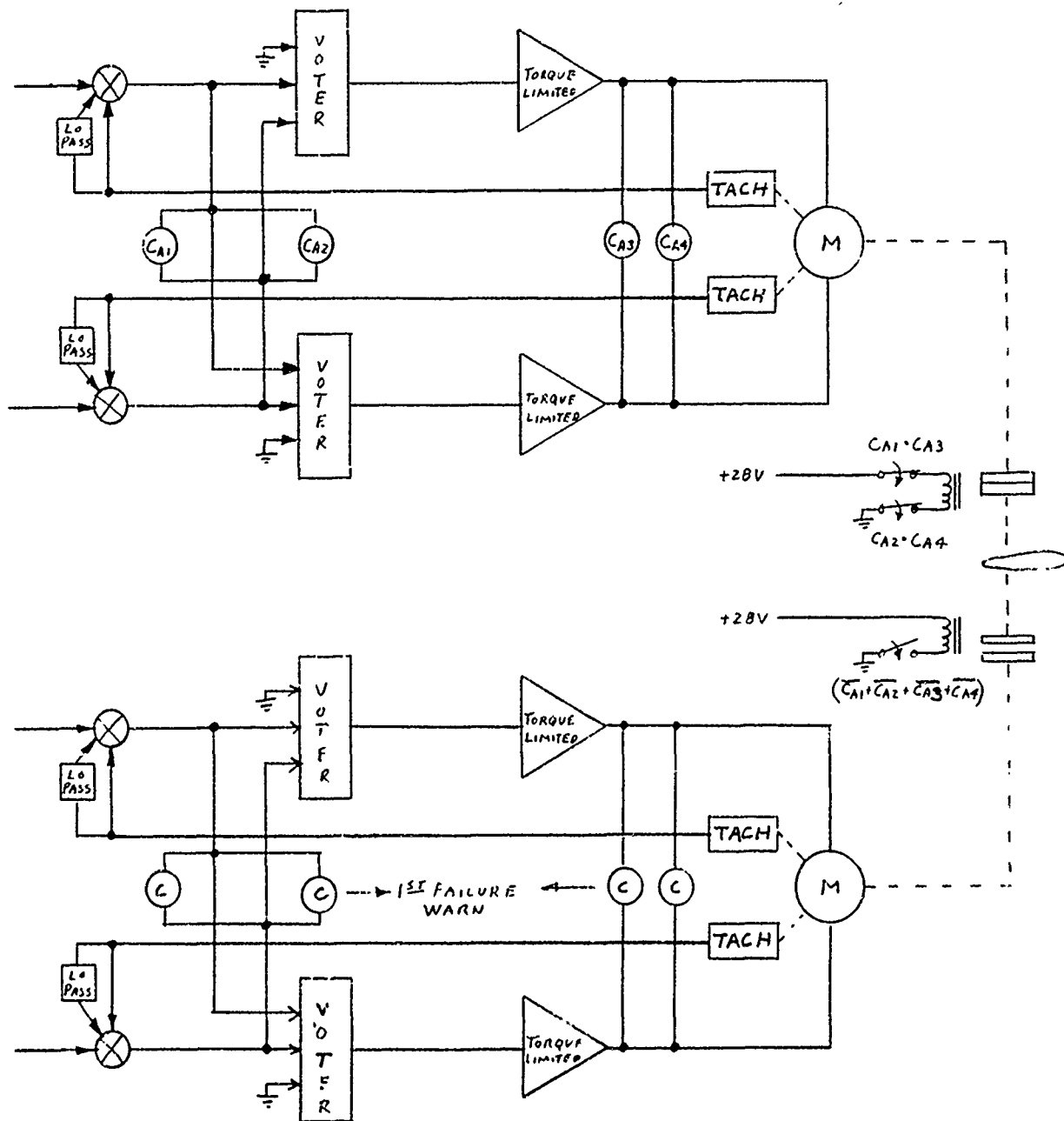
Both techniques have deficiencies, but the alternate-engage technique is preferred because of its lower cost. Crossfeeding comparator logic to drive the other-side clutch imposes safety problems in the alternate-engage configuration. Single failures must be precluded from prematurely engaging servo 2 while servo 1 is engaged, thereby causing a fighting situation and an eventual double servo disconnect. Similarly, care must be taken in the torque-summing configuration to preclude crossfeeding of failures via the equalization crossfeeds.

The alternate-engage configuration does not have voters and a mechanical differential and, consequently, is less expensive.

8.1.4

Sensor Interface

Fault analysis in Appendix A shows that crossfeeding of sensor outputs into the flight computers is required, except in the case of dual sensors into dual flight computers. It is further shown that only in the case of triple sensors driving triple computers should sensor data be processed and crossfed between computers in digital form. In all other cases, raw sensor data crossfeeding is preferred.



DUAL SERVO CONFIGURATION
WITH
ALTERNATE-ENGAGE SWITCHING
FIG. B.4

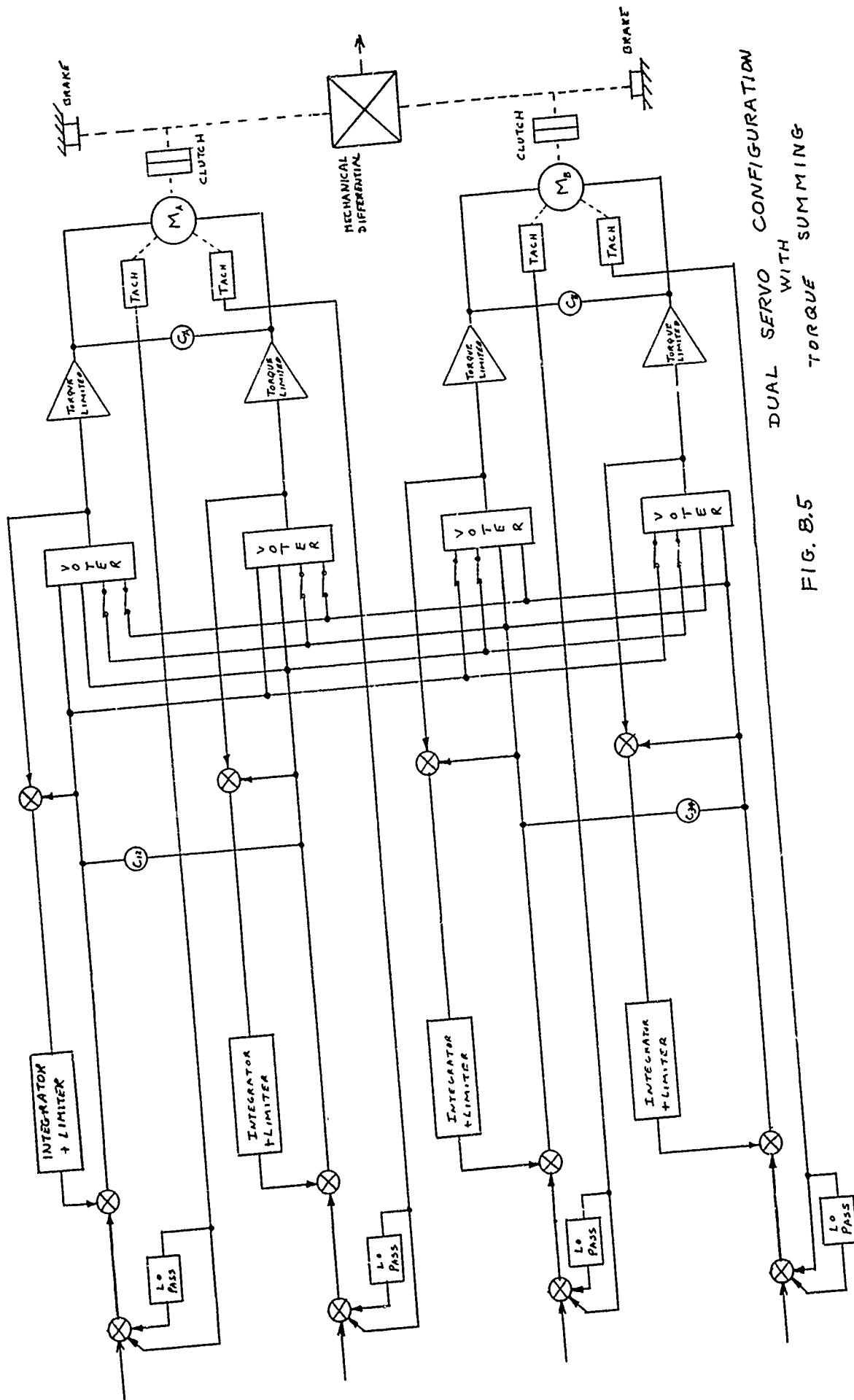


FIG. B.5
DUAL SERVO CONFIGURATION
WITH TORQUE SUMMING

8.1.5

Servo Interface & Equalization

A fault analysis can show that crossfeeding of servo commands is required in a triplex system and desirable, but not required, in a dual-computer system. As discussed in Appendix A, however, equalization considerations force crossfeeding in the dual. Once the need for crossfeeding has been established, voting is shown to be a preferred technique. In the triplex system, voting must be implemented downstream of the DFC's in analog hardware located in the servo amplifier LRU's. In the dual systems, voting can be more economically implemented in the DFC's in software.

8.1.6

Independent Red-Line Monitor

An independent device which monitors aircraft performance can detect the effects of undetected failures before vehicle performance deteriorates excessively. An independent red line monitor box was defined for those configuration candidates which do not contain red-line monitors within their flight computers. A 3/8 ATR short package weighing 8 pounds was estimated for the monitor.

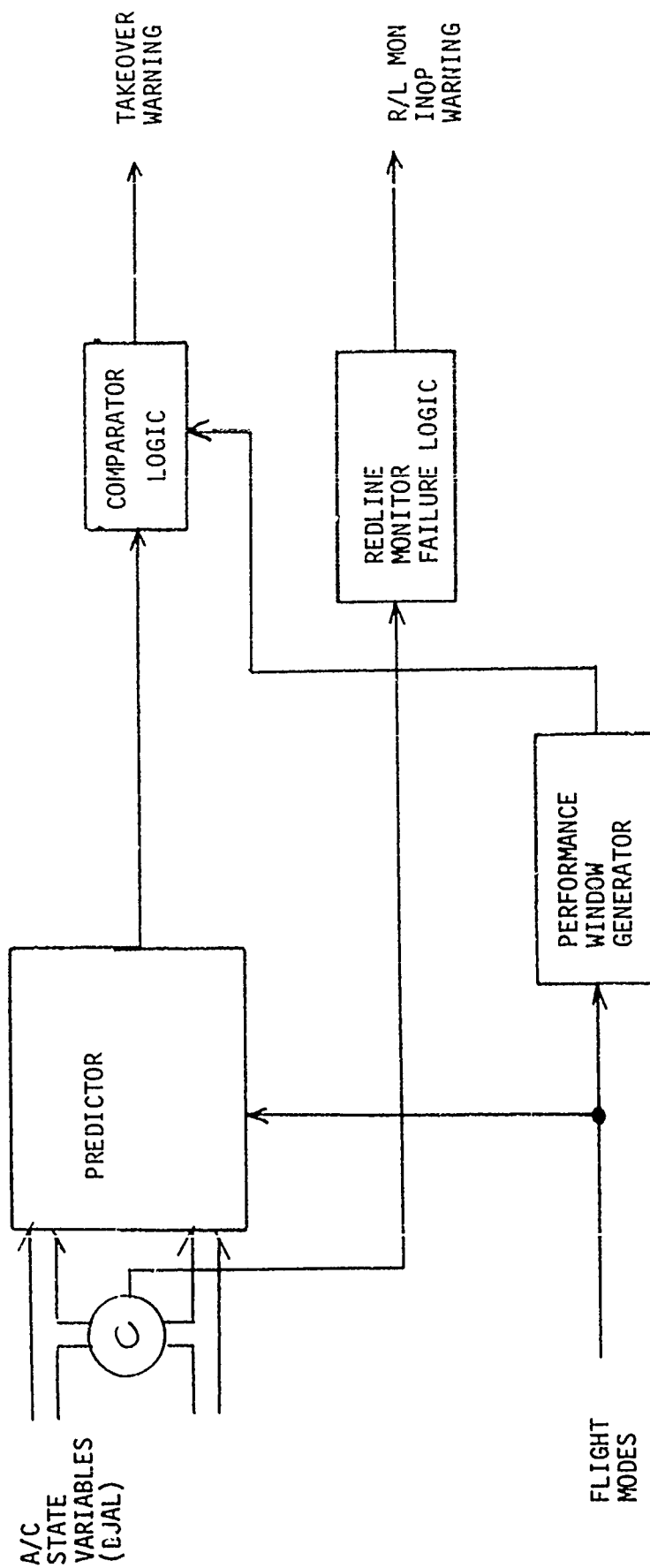
A suggested functional diagram is shown in Fig. 8.6. The predictor continually predicts vehicle performance based on current vehicle states. In cruise, for example, attitude rates can be used to predict potentially-excessive vehicle attitudes. In recovery, MLS and radar-altitude information can be added to the prediction algorithm to predict vehicle touchdown point. A performance-window generator provides a set of acceptable performance values which are compared with the predicted set. An out-of-window prediction alerts the RO and automatically switches the appropriate servos to a backup autopilot computer.

It was felt that a microprocessor would be the most cost effective implementation, provided monitor sophistication remains relatively low. Recovery mode will be the most demanding. The closer to touchdown the monitor is expected to perform, the more sophisticated the prediction algorithm, since the normal "tightening-up" of the landing maneuver makes it harder to distinguish nominal from abnormal performance.

One might argue that implementing a red-line monitor with a digital processor makes the monitor susceptible to all of the same digital processor anomalies it is expected to detect. A red-line monitor will detect the effects of:

- a. Undetected DFC hardware failures
- b. Software algorithm problems

The monitor can detect DFC hardware failures no matter how implemented, since it monitors vehicle performance. Since the microprocessor can be a different machine than the DFC CPU and will contain an altogether different software package, software problems will not occur simultaneously in both processors. Thus it is felt that a microprocessor implementation is quite safe.



REDLINE MONITOR
FUNCTIONAL DIAGRAM

FIG. 8.6

Because of the system importance of a red line monitor, the red-line monitor, itself, should be well monitored. For this reason, dual sets of sensor inputs are desirable, as shown in Fig. 8.6. A failure of the red-line monitor must immediately alert the RO.

8.1.7 Backup Analog Flight Computer

A manual-mode backup analog flight computer function is shown in the redundancy study fault analysis to be a requirement. For those configurations which do not contain well-monitored DFC's with internal backup autopilot computations, an independent backup must be provided. A minimum analog backup computer has been defined and could be packaged in a 10-pound 1/2 ATR short box. The computer would provide pitch, roll, yaw, and throttle channels of computations.

A block diagram of a backup flight computer is shown in Fig. 8.7. Sophisticated modes, like autoland, are not required. Only those modes necessary to allow the RO to remotely fly and land the vehicle are provided. These include:

- a. Control-stick steering with data link tie-in. Vehicle attitude is maintained until modified remotely by the RO. The computer can accept pitch and roll commands from either up link.
- b. Yaw SAS with turn coordination. Yaw rate damping with a aileron-to-rudder coordination feed. In runway alignment a heading-hold mode will, most likely, be required, which will thus require a heading input from the compass system and an up link tie-in.
- c. Throttle. Either remote manual throttle position or throttle-rate command via up link.
- d. Go-Around. Fixed pitch command in pitch and wings-level in roll.

More exotic modes could be added to minimize RO workload during manual takeover, space permitting, within the backup computer.

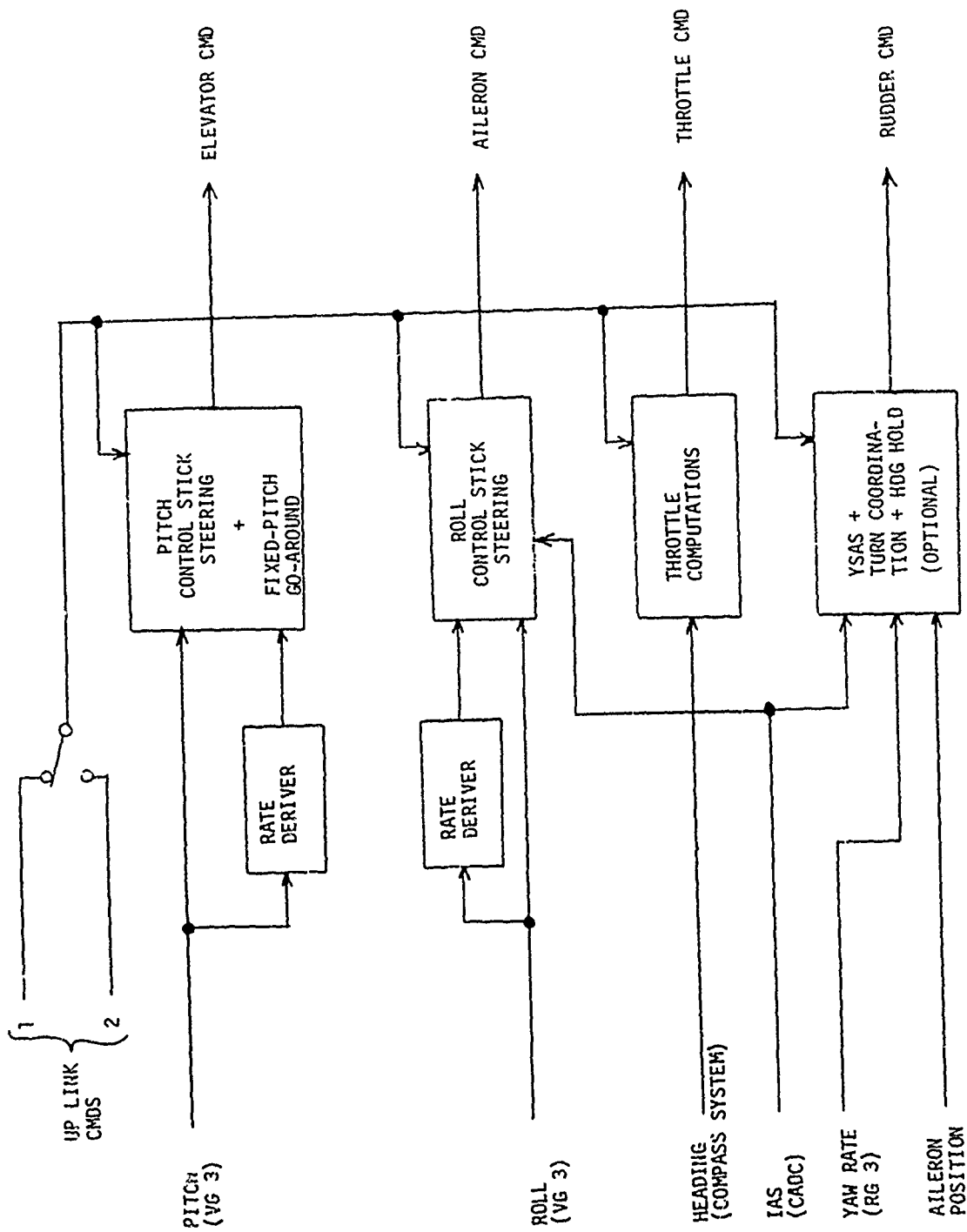
8.2 Definition of Candidates

Starting with a fail-operational set of sensors and servos (Section 8.1), a progression of configuration candidates can be generated in order of increasing complexity by adding flight computers and monitoring.

8.2.1 Configuration A

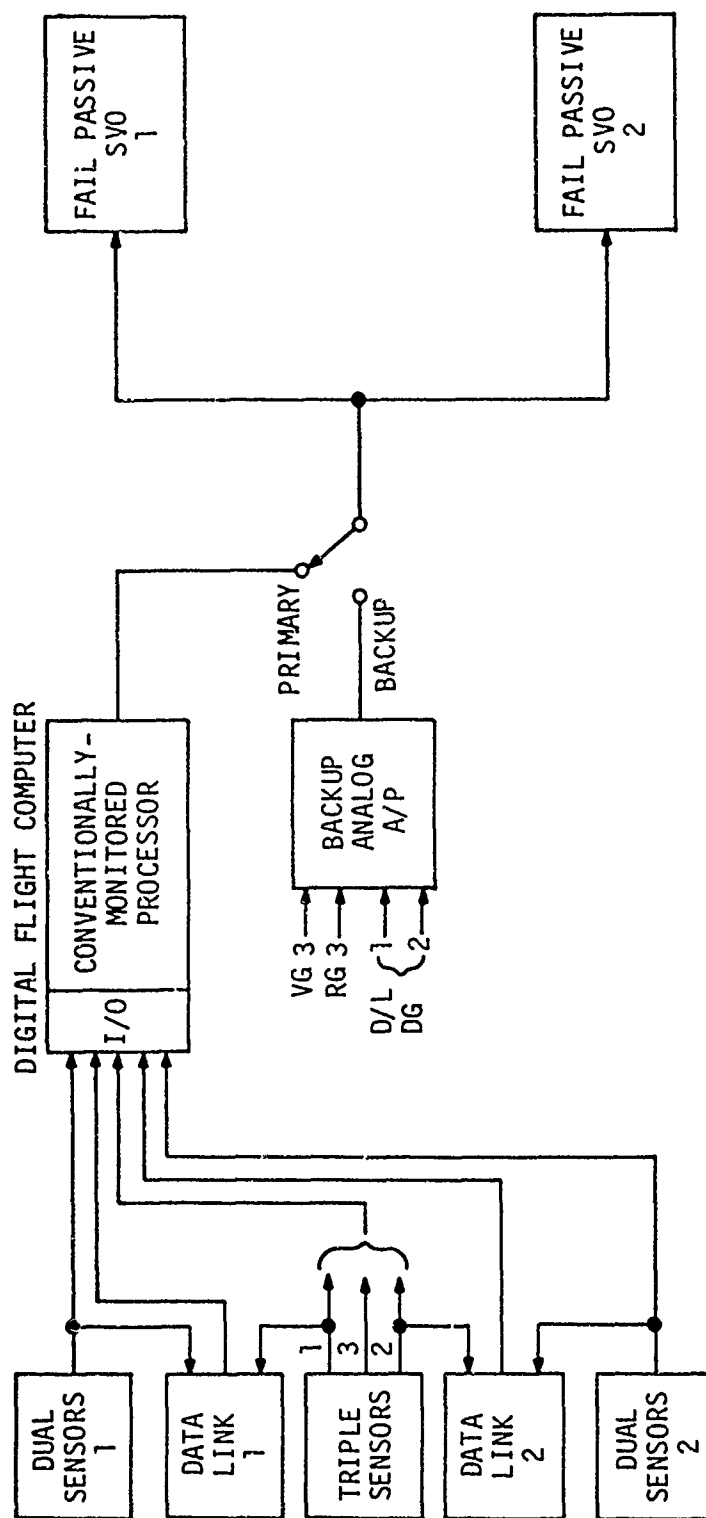
The simplest system imaginable is a single conventionally-monitored DFC with a single backup analog computer, configuration 8.8. The set of sensors are voted within the DFC. No red-line monitoring is provided. Reversion to the backup autopilot computer occurs solely on command of the DFC in-line monitor. Since the monitoring level is estimated to be only 75%, 25% of DFC hardware failures and software problems are transmitted directly undetected to the servos.

The backup autopilot computer is fed from the middle vertical gyro and rate gyro and both up-link systems.



BACKUP ANALOG FLIGHT COMPUTER
FIG. 8.7

FIG. 8.8
CONFIGURATION
A



8.2.2 Configuration B

If an independent red-line monitor is added to Configuration A to catch these otherwise undetected hardware and possible software problems, Configuration B results, 8.9. Reversion to the backup now occurs on command of either the DFC in-line monitor or the red-line monitor.

The small cost and weight penalty increases the system totals to \$219K and 378 lbs, respectively.

Sensor interfacing with the down links can be provided by the DFC and red-line monitor, the #1 sensors interfacing via the DFC and #2 sensors via the red-line monitor.

8.2.3 Configuration C

Intuitively, Configurations A and B would seem to be deficient in takeoff and recovery phases, where manual takeover is often unsuccessful and higher computer redundancy would be desirable to provide additional automatic capability. Accordingly, a second modestly-monitored DFC is added to Configuration B to yield Configuration C, 8.10. Total system cost and weights for this configuration are estimated to be \$279K and 413 lbs, respectively.

In this configuration, only red-line monitor trips cause reversion to the backup computer. The DFC in-line monitors cause output voter reconfiguration and remove the failed DFC output from the servo inputs.

The available symmetry permits each DFC to interface its-side sensors with its-side data link. Maximum use is made of the free-wheeling I/O within each DFC to minimize dependence of down-link telemetry on DFC status.

8.2.4 Configuration D

If highly-monitored DFC's are substituted for the modestly-monitored DFC's of Configuration C, the independent red-line monitor and backup autopilot computer may be combined within the DFC's internal microprocessor, as discussed above. This configuration of dual highly-monitored DFC's is defined as Configuration D, 8.11. The system cost and weight for this configuration are estimated to be \$269K and 395 lbs, respectively, a reduction of \$10K and 18 lbs, respectively, from Configuration C.

The servo command outputs are voted in software within each DFC before outputting to the servos, as in Configuration C. The required servo command crossfeeds can be digital. Down-link interfacing is identical to Configuration C.

FIG. 8.9
CONFIGURATION
B

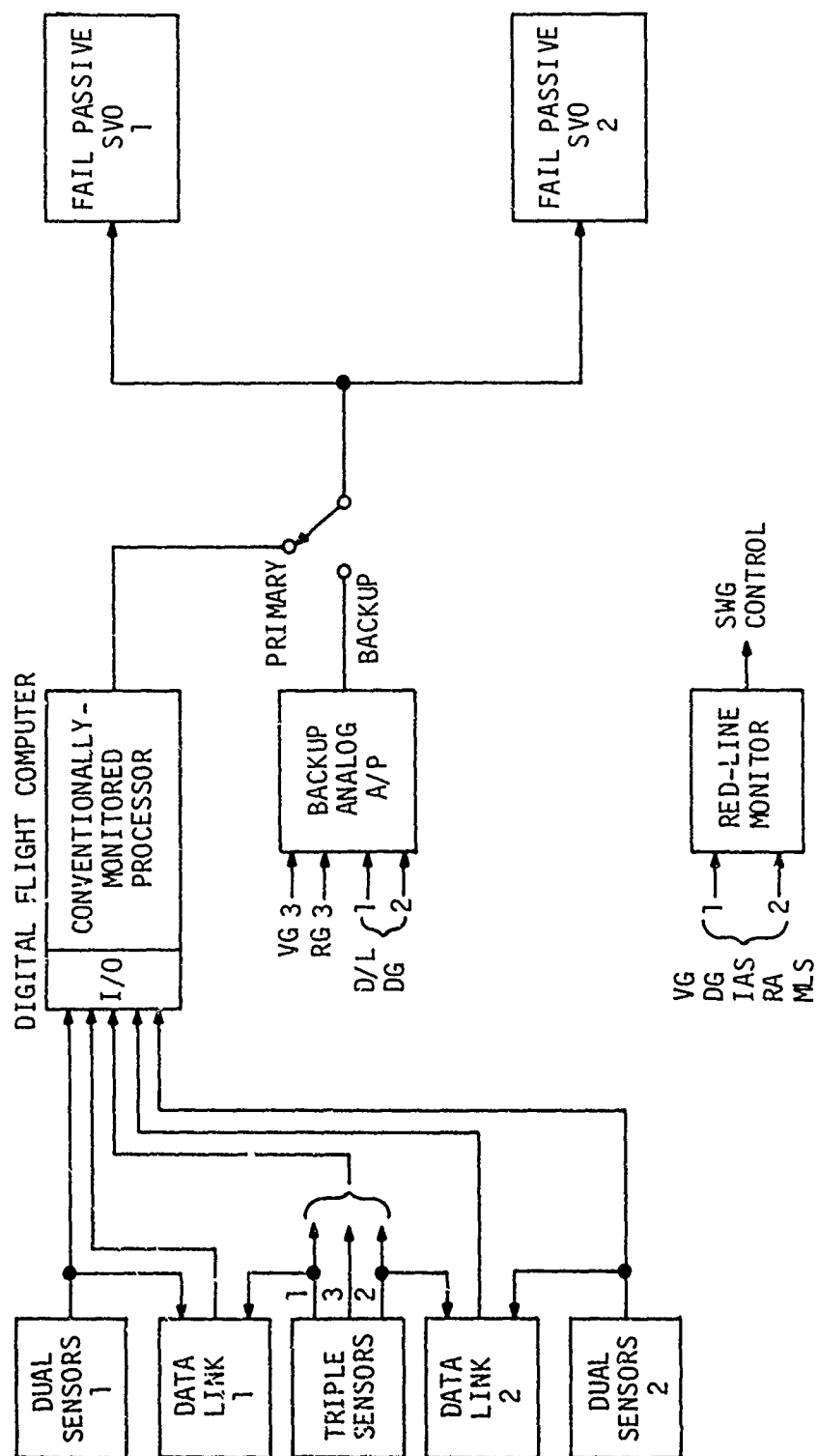


FIG. 8.10
CONFIGURATION
C

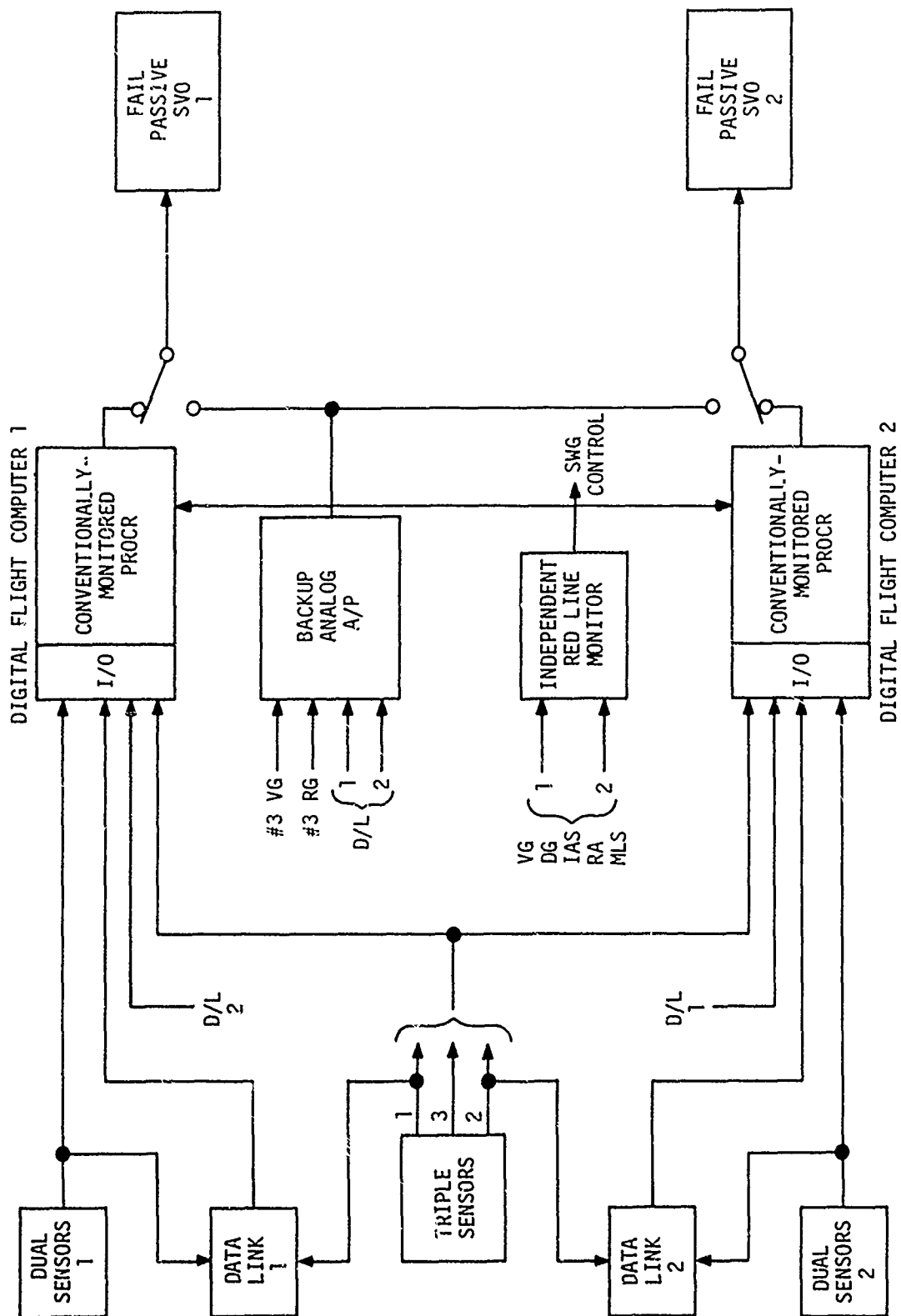
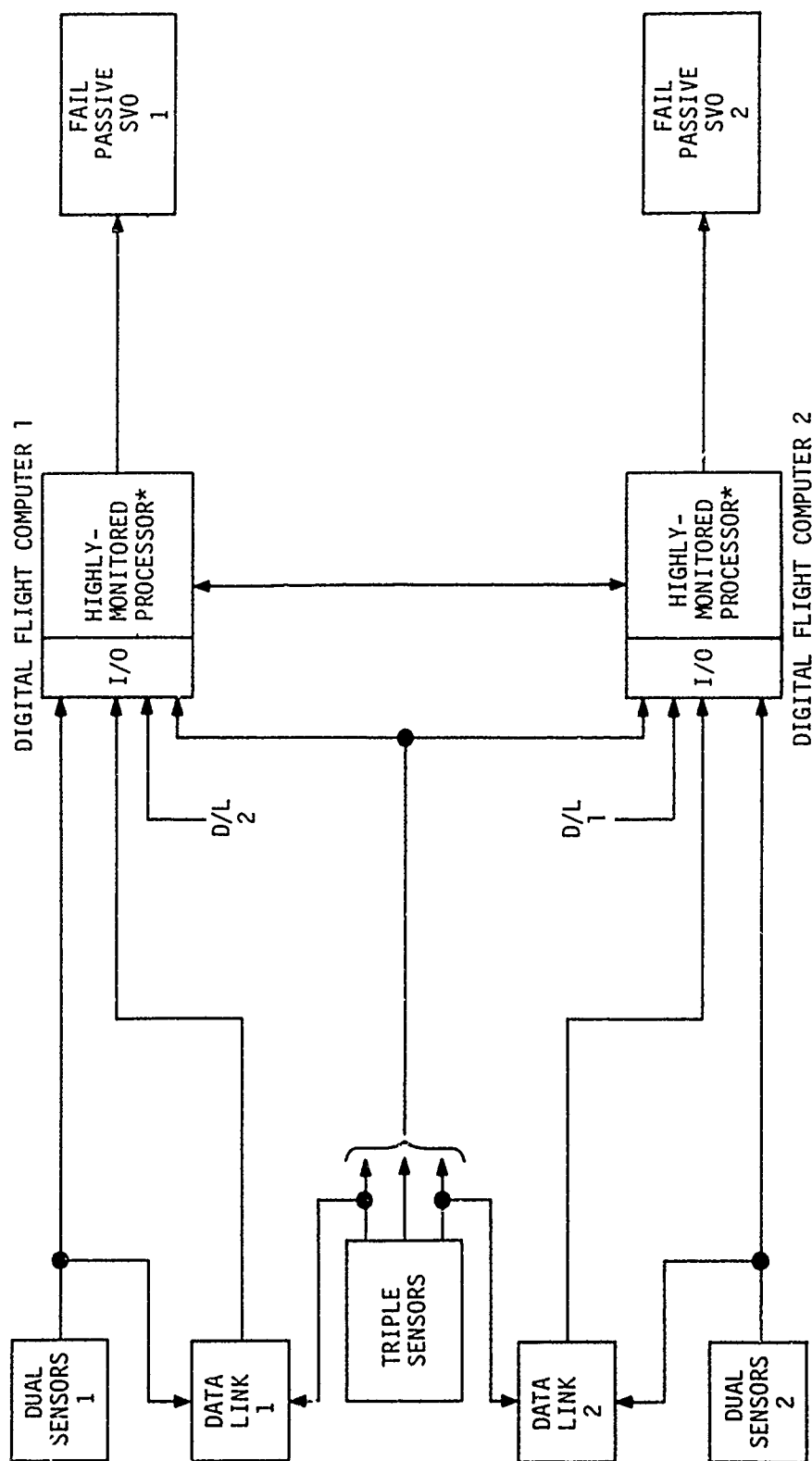


FIG. 8.11
CONFIGURATION D



*INTERNAL MICROPROCESSOR PROVIDES:

RED-LINE MONITOR
BACKUP A/P FUNCTION

8.2.5

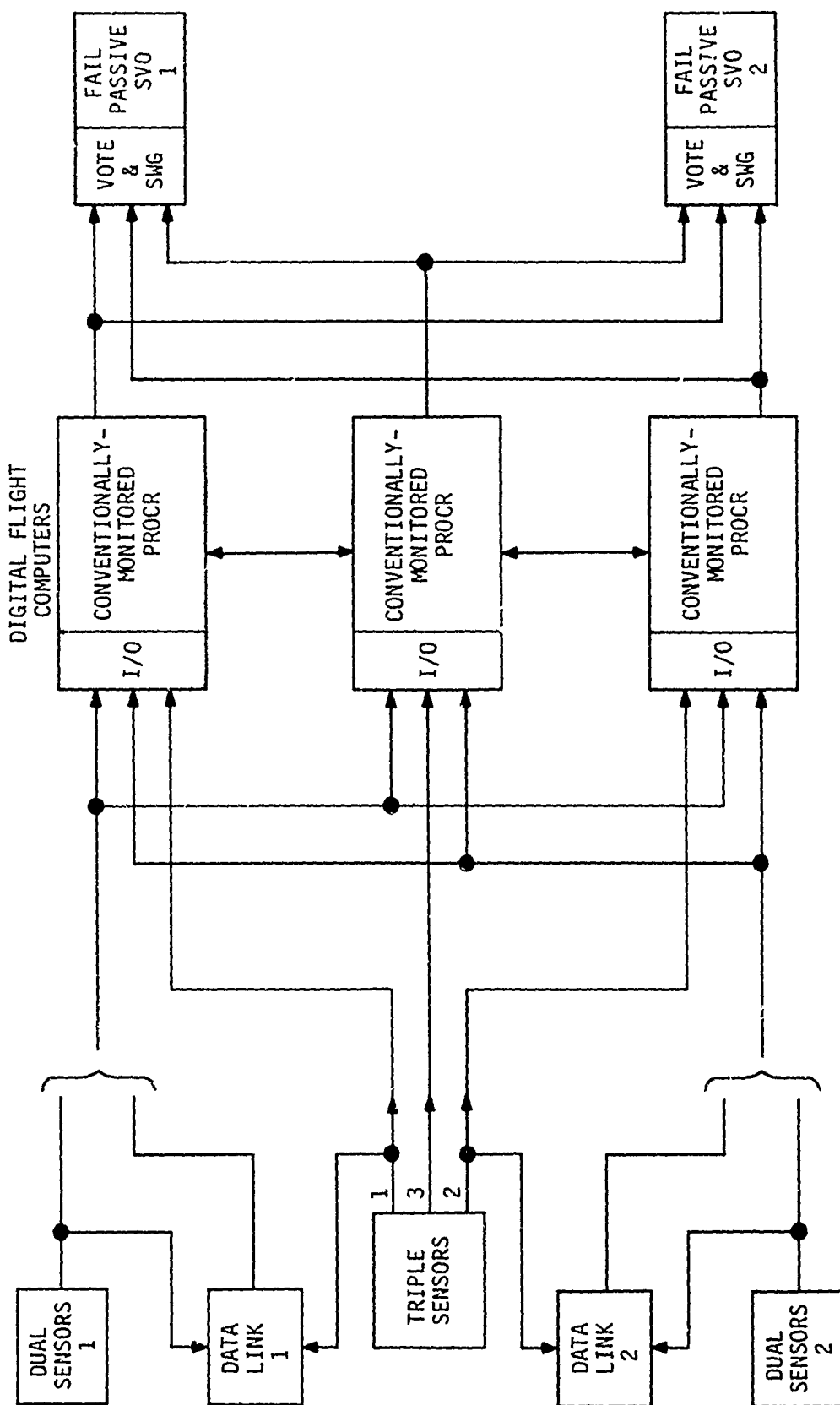
Configuration E

Configuration E, 8.12, is a triplex system built around three conventionally-monitored DFC's. It is not a pure-triplex configuration, since the servos and several sensors are dual in-line monitored. It approximates, however, the triplex FCS configuration, currently being proposed for transport aircraft autoland application. No backup analog computers or red-line monitors exist in Configuration E. E provides a means of evaluating the reliability of a transport triplex configuration. The cost and weight of E are \$323K and 430 lbs, respectively, the most expensive and heaviest of the five candidates.

The outputs of the three DFC's are crossfed, compared, and voted in analog form downstream in the servos. This configuration, however, allows all DFC hardware failures to be detected - a significant advantage over the dual-computer configurations, where only up to 95% of the hardware failures are detectable by in-line monitoring. It, conversely, reduces the monitoring-level requirements on its DFC's. This configuration, however, cannot accommodate software problems, should they occur. A red-line monitor and backup computer would have to be added if software was considered a potential failure mode.

Down-link interfacing is provided by the two outside DFC's. Triplex sensors are crossfed between DFC's in digital form to reduce I/O requirements.

FIG. 8.12
CONFIGURATION
E



9.0 FAULT ANALYSIS RESULTS

This section discusses the results of the failure analysis. The details of the fault analysis, including the fault trees and the probability equations, are included in Appendix B.

9.1 Fault Analysis Summary

The definitions of the system configurations that were used in this study were given in Section 9. Table 9.1 contains a cost/weight/probability comparison chart. The hardware costs were tabulated from the unit prices given in Table 8.1 and the hardware units required for each configuration.

The data for the probability of a vehicle loss due to an FCS failure is taken from the results of the calculations in Appendix B. For the sake of brevity in the ensuing discussion, the term probability of vehicle loss is meant to mean the probability of vehicle loss due to an FCS failure (or a combination of FCS failures).

The probability of vehicle loss for the entire flight given in Table 9.1 is almost totally due to the probability of loss during the cruise phase of the mission. Except for configuration A, the probabilities of loss for the entire flight fall within 5% (actually 3.5%) of 0.0020. The significance of this is discussed in the following section.

The probability-of-loss values for takeoff and for recovery tend to follow the redundancy and monitoring levels of the various configurations. There is a definite observable variation in the probability-of-loss values.

The similarity of the entire-flight-loss probability values does not give any basis of selection between the various systems. It is recommended that the takeoff and recovery data be used for comparison purposes. The control system configuration does play an important part in their makeup. If a recommendation were to be made based on the data in Table 9.1, it would be either configuration B or configuration D. The final selection would depend upon the emphasis placed on the recovery probability. Though costing 25% more than configuration B, configuration D exhibits an order-of-magnitude better recovery reliability.

9.2 FCS Probability-of-Loss Discussion

The data link, sensors, and servos for all of the candidate configurations are the same. Only the digital flight computer redundancy, the redline monitors, and the backup hardware change. The extent to which the DFC's are self monitored also enters in the probability-of-loss considerations.

As stated above, the probability of vehicle loss for the entire flight is due almost exclusively to the cruise loss. The takeoff and recovery loss probabilities add less than 0.2% to the entire-flight-loss probability.

In the fault analysis of appendix B, the data link probability includes the loss of the flight critical sensors. Besides being required for automatic flight control, the remote operator requires these sensors for assessing vehicle operation. The sensors are not crossfed to the down link. If one flight-critical sensor and the opposite data link fails, the remote operator

SYSTEM CONFIGURATION	COSTS		WEIGHT	FCS PROBABILITY			MISSION ABORT
				VEHICLE LOSS			
	HARDWARE	DESIGN		ENTIRE FLIGHT	TAKEOFF	RECOVERY	
A	(K dollars) 210	(K dollars) 1329	(lbs) 370	0.0035	($\times 10^{-5}$) 0.556	($\times 10^{-6}$) 4.22	0.133
B	219	1896	378	0.00203	0.556	2.62	0.137
C	279	1896	413	0.00198	1.04	0.888	0.145
D	273	1008	395	0.00193	0.236	0.214	0.139
E	323	958	430	0.00203	0.05	0.059	0.144

SYSTEM COST/RELIABILITY COMPARISON

IDEAL SOFTWARE ASSUMED

TABLE 9.1

SYSTEM CONFIGURATION	COSTS		WEIGHT (lbs)	FCS PROBABILITY			
	HARDWARE (K dollars)	DESIGN (K dollars)		VEHICLE LOSS			
				ENTIRE FLIGHT	TAKEOFF	RECOVERY	
					(x 10 ⁻⁶)	(x 10 ⁻⁶)	MISSION ABORT
A	210	1329	370	0.00854	2.26	12.8	0.133
B	219	1896	378	0.00213	2.26	5.83	0.137
C	279	1896	413	0.00208	2.74	2.32	0.145
D	273	1008	395	0.00193	1.94	1.64	0.139
E	323	958	430	0.00707	1.75	8.64	0.144

SYSTEM COST/RELIABILITY COMPARISON

WITH UNRELIABLE SOFTWARE

TABLE 9.2

Software Assumptions

The following extremely pessimistic assumptions were made on software reliability:

1. Software algorithm problems occur once every 5000 hours.
2. All software problems have catastrophic effects.

does not have the data necessary to monitor the flight. In this case it has been assumed that ultimately the vehicle would be lost. Sensor failures that degrade down-link monitoring by the remote operator are included in the data link system failure rates.

For all the configurations, except for configuration A, the data link and sensors in cruise account for more than 94% of the vehicle losses. The loss probability due to these items alone is 0.00191 failures/flight. To show the effect of the data link and the vertical gyro on the probability of loss, a data link with a 2.5:1 improved MTBF and a vertical gyro with a 2:1 improved MTBF were considered. Both of these improvements are possible within the state of the art for these devices, although there may be a cost and weight penalty. With these improvements, the probability of loss due to the data link and sensors is 0.00047 failures/flight. This is a 4:1 improvement in the cruise loss probability. Improved data link and sensor reliability should, therefore, definitely be a subject area for further study.

9.3 In-Line Monitoring of a Digital Processor

In a system fault analysis, processor hardware and software can be treated as two independent failure sources. Undetected failures in either can be extremely serious in a flight control system. Cross-channel monitoring downstream of the processors can catch hardware failures not detected by the in-line monitors, but software bugs generally defy detection.

A 75% level of in-line monitoring is achievable in a digital flight computer essentially for free. A relatively high level of 95% can be achieved at small extra cost. Assurance of perfect monitoring, however, is limited by the ability to anticipate all possible failure modes. If a microprocessor is used to perform the monitoring function within a digital flight computer, its excess capability may be used for red-line monitor and simple backup autopilot functions.

9.4 In-Line vs. Red-line Monitoring

Several important differences between in-line and red-line monitoring techniques are worth noting:

- a. A high-level in-line monitor can detect hardware failures closer to touchdown in the recovery phase than a red-line monitor.
- b. A red-line monitor, in many cases, can catch the effects of software bugs, whereas an in-line monitor cannot.
- c. A red-line monitor, in many cases, can detect hardware failures not detected by in-line monitoring.

9.5 Effect of Software Reliability

Software failures were considered in this study. Their effect is shown in the data in Appendix B. In a multiple-processor system, a software problem can affect all processors at the same time if the same programs are used. Systems that utilize a redline monitor are not as susceptible to such failures because the redline monitor is assumed to have separate and independent software. Table 9.2 shows the effect of such software failures. For the calculations, a failure rate of 210×10^{-6} failures/hour was used for the software.

This is probably not a realistic value, but it does show the susceptibility of the various configurations to such failures. Comparison of Tables 9.2 and 9.3 shows, as would be expected, that configurations A and E which do not have redline monitors are most affected by the software failure. It is felt that, although software failures will probably not be as bad as used in the example, this is a potential problem area that should not be overlooked in future considerations.

9.6 Probability of Mission Abort Discussion

The mission abort probabilities were calculated over the period from takeoff through cruise. The exposure time used is 23 hours. A failure during the last hour of the mission was not considered to be an abort.

A mission abort was considered to be necessary if any sensor, servo, data link or digital flight computer (DFC) unit fails. The combined failure (to cause an abort) probabilities of the sensors, servos and data links is 0.1270 aborts-due-to-failures/hour and is the same for all five configurations. The DFC units of the various configurations only increase this probability by 14% or less. In fact, sensor failures account for 45 to 50 percent of the mission aborts. The data link failures account for 32 to 35% of the mission aborts. It is obvious that in order to improve the mission abort probability, the sensors and data link are prime areas for improvement.

Using a more reliable vertical gyro (250×10^{-6} failures/hour) and data link (400×10^{-6} failures/hour) improve the mission abort probability by about 30%.

All the mission abort probabilities tabulated in Table 9.1 exceed the target probability of 0.034 aborts/flight. Another non-trivial assumption that makes the abort probability as good as it is that the MLS receivers and the radio altimeters are not powered above 10,000 feet MSL. This reduces the exposure time of these units. More importantly, the failure rate of these is much greater if they are required to operate in ambient conditions above 10,000 feet.

10.0 RECOMMENDATIONS FOR FURTHER STUDY

Several topics are suggested for further study. These topics either became apparant during the course of the study or were treated superficially because of time limitations. The topics include:

1. Software-Reliability Assessment - The redundancy study has shown the susuptibility of certain configurations to software "bugs" which defy detection by normal in-line monitoring techniques. Very little software reliability data is available in the literature. The data that was found and used to test system software error susceptibility is felt to be exceedingly pessimistic. Consequently, to derive loss rates that are quantitatively meaningful, a software reliability program should be initiated. From the data a more meaningful software MTBF could be derived.
2. Sensor/Data - Link Reliability Improvement - The study has shown that the probability of vehicle loss for the Compass Cope is high and very sensitive to sensor and data-link system reliabilities. Of the sensors, the vertical gyros are the most offensive with their low MTBF of 2000 hours. A study should be undertaken to explore substituting a hi-grade commercial gyro with its typically higher reliability, for the military 9000-C gyro. Similarly, a data link reliability improvement program should be conducted.
3. Sensor Simplification - A study should be initiated to determine if the computing capability of the digital flight computers could simlify the sensor requirements. For example, some of the functions of the air data computers might be performed within the DFC's, permitting less-expensive air data computers.
4. Hydraulic ServoInvestigation - Time did not permit consideration of both electro-mechanical and hydraulic control-surface servos, though the need for fail-operative servo redundancy was established. Consequently, a search for potentially-suitable hydraulic actuators should be initiated, followed by a tradeoff study with electro-mechanical servos.
5. Mission Abort Improvement - The probabilities of mission abort for all of the redundancy configurations are high and exceed the suggested target values. Baring a marked improvement in individual system component reliabilities, it would appear that a significantly lower mission-abort rate for the Compass Cope FCS is inconsistent with the redundancy levels required. Given the unusually long mission duration of Cope, a redefinition of the normal mission abort groundrules might be in order. For example, is it reasonable to score as an abort a vertical gyro failure in the 20th hour of a 24- hour mission? For safety reasons, however, such a failure should scrap a mission, since FCS redundancy would then be less than nominal.

- 7-1
6. Investigate Effects of Less-than-Perfect Ground and In-Flight Verification Tests - To facilitate the fault analysis perfect (100%) pre-flight verification of the entire FCS were assumed to preclude latent faults within the system prior to takeoff and recovery phases. This is unrealistic, and the fault analysis ought to be modified to reflect realistic test levels.
 7. Quantify the Civil Airspace Hazard - Vehicle loss rates, alone, do not assess the potential damage to life and property apart from the vehicle itself. A method for deriving hazard probabilities was discussed in Section 3.0. No quantitative data suitable for defining a hazard requirement was found. It is suggested, therefore, that a study be initiated to obtain such data.

11.0 REFERENCES

1. Goldberg, J. and Wensley, J. H., "A Forward View on Reliable Computers for Flight Control," Computer Science Group, Stanford Research Group, Menlo Park, Calif.
2. Hendrick, R. C. and Hill, C. D., "Self-Testing Digital Flight Control Applications," Honeywell Inc., Minneapolis, Minnesota (1975)
3. Osder, Stephen, "Architecture Considerations for Digital Automatic Flight Control Systems," presented at ARINC Avionics Engineering Seminar, May 23, 1975.
4. Osder, Stephen, "The Implementation of Fail-Operative Functions in Integrated Digital Avionics Systems," NASA Advanced Control Technology, July 9-11, 1974.
5. Raymond, R. G., and Larson, J. C., "Digital Computation Makes AFCS More Reliable," Honeywell, Inc., Minneapolis, Minn.
6. Yopp, W. P., and McDonnell, J. D., "Digital Flight Control Systems - Considerations in Implementation and Acceptance," AIAA Digital Avionics Systems Conference, Boston, Mass., April 2-4, 1975.
7. John McGough, et al, "Digital Flight Control System Redundancy Study," Air Force Flight Dynamics Laboratory Report, AFFDL-TR-74-83, July 1974.
8. R. C. Hendrick, et al, "Design Criteria for High-Authority Closed-Loop Primary Flight Control Systems," Air Force Flight Dynamics Laboratory Report, AFFDL-TR-71-78, August 1972.
9. J. C. Hall, et al, "Digital Flight Control: An Approach to Efficient Design," IEEE Transactions on Aerospace & Electronic Systems, Vol. AES-11, No. 5, Sept. 1975.
10. J. H. Boone, et al, "Digital Automatic Flight Control is the Answer - Now, What's the Question?," IEEE Transactions on Aerospace & Electronic Systems, Vol. AES-11, No. 5, Sept. 1975.
11. "Annual Review of Aircraft Accident Data, U. S. General Aviation, Calendar Year 1973," National Transportation Safety Board, NTSB-ARG-75-1, 25 July 1975.
12. "Annual Review of Aircraft Accident Data, U. S. Air Carrier Operations, 1973," National Transportation Safety Board, NTSB-ARC-74-2, 24 Oct. 1974.
13. "Air Force Aircraft Safety Records," 1973.
14. 1975 World Almanac.
15. "Formulation of Preliminary Control Laws for Autoland of Compass Cope Remotely Piloted Vehicles, Final Report," by Collins Radio, AFFDL/FGC Contract No. S-74-9, 22 August 1974.
16. A. M. Mood and F. A. Graybill, "Introduction to the Theory of Statistics," New York, McGraw-Hill, 1963.

11.0 REFERENCES (Continued)

17. I. Miyamoto, "Software Reliability in On-Line Real Time Environment,"
International Conference on Reliable Software.

APPENDIX A
SENSOR & SERVO REDUNDANCY REQUIREMENTS

NEED FOR FAIL-OPERATIVE SENSORS

A single complement of sensors is inadequate. Consider a group of sensors necessary for cruise: a vertical gyro, a CADC and a rate gyro. The combined failure rate for these sensors is 7.95×10^{-4} failures/hour. Thus the probability of a sensor failure during a 24 hour flight is 0.0191 failures per flight, or almost 2 failures in 100 flights. This rate of failure is not acceptable. Using dual unmonitored sensors provides a fail-soft capability, but does not improve (and actually degrades) the system reliability. A sensor disagreement can be detected, but which of the two sensors has failed cannot be easily determined and re-engagement of the failed sensor could be catastrophic.

In the case of inline monitored sensors, only two are required because the inline monitoring flags the sensor that has failed. For unmonitored sensors, three are necessary before any failure rate improvement benefits are effected. With three sensors it takes two sensor failures before the system is not operable.

With the above sensor complement (assuming triple sensors) the failure rate for a 2 out of 3 failure is 6.32×10^{-7} failures/hour. The probability of this occurring during a 24 hour flight is 0.000364 failures per flight. This is an acceptable failure rate, whereas the 0.0191 failure rate for single sensors is not acceptable.

JUSTIFICATION FOR SENSOR CROSSFEEDING

General

For the following discussions assume that the dual sensors are 100% inline monitored and the triplex sensors are not monitored. Consider the system configurations in the following table:

Configuration Number	<u>System Configurations</u>	
	<u>Sensor Redundancy</u>	<u>Computation Channels</u>
1	2	1
2	2	2
3	2	3
4	3	1
5	3	2
6	3	3

From a topological point of view it is obvious that the sensors in configuration 1 and 4 must be crossfed into the one computation channel in order to be utilized. In configuration 3, crossfeeding of the sensors is necessary for the third channel. To keep the computation channels all the same for this configuration, it is necessary to crossfeed the sensors to the other channels as well. This keeps the different versions of the hardware units to a minimum. Thus it remains to show that configurations 2, 5, and 6 require crossfeeding.

For the following discussion use a combination of several sensors (vertical gyro, rate gyro, CADC) to form a hypothetical sensor. See the section on the "Need for Fail-Operative Sensors" for more detail. Let the failure rate for this hypothetical sensor be λ_s , where

$$\lambda_s = 7.95 \times 10^{-4} \text{ failures/hour}$$

Assume the failure rate for the computation channel to be

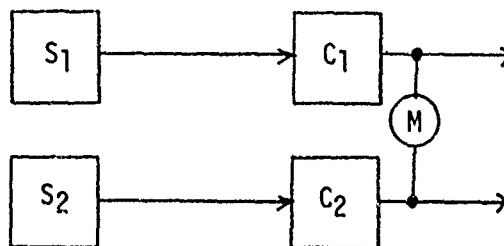
$$\lambda_c = 2.50 \times 10^{-4} \text{ failures/hour}$$

Let the exposure time, T, be 24 hours.

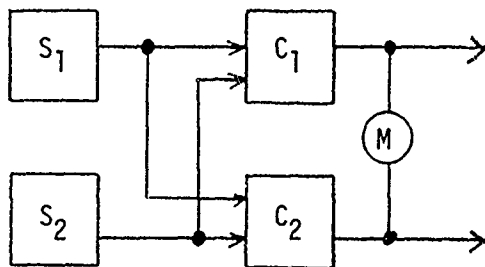
It is also assumed that outputs of the computations (input to the servo amplifiers) are cross channel monitored.

CONFIGURATION 2

For configuration 2 there are two possible ways to connect the sensors.



Configuration 2a



Configuration 2b

The probability of a total system failure for Configuration 2a can be written as:

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P[S_1] P[S_2] + P[C_1] P[C_2] + P[S_1] P[C_2] \\
 &\quad + P[S_2] P[C_1] \\
 &= \lambda_s^2 \tau^2 + \lambda_c^2 \tau^2 + 2 \lambda_s \lambda_c \tau^2 \\
 &= 6.29 \times 10^{-4} \quad \text{per flight}
 \end{aligned}$$

The cross feeding of the sensors in Configuration 2b eliminates the cross channel failures due to one sensor and the opposite computation channel. The probability of failure for Configuration 2b is:

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P^2[S] + P^2[C] \\
 &= 4 \times 10^{-4} \quad \text{per flight}
 \end{aligned}$$

The failure rate for Configuration 2b is a 38% improvement over that for 2a. Sensor crossfeeding is an obvious benefit to this configuration in cruise, but not a requirement.

Repeating the analysis for the recovery phase, the probability of a total system failure for Configuration 2a similarly becomes:

$$P[\text{TOTAL FAILURE}] = \lambda_s^2 \tau^2 + \lambda_c^2 \tau^2 + 2 \lambda_s \lambda_c \tau^2$$

$$P[\text{TOTAL FAILURE}] = 0.0024 \times 10^{-6}$$

$$\text{if } \lambda_s = \lambda_{RA} + \lambda_{MES} = 958 \times 10^{-6} \text{ hr}^{-1}$$

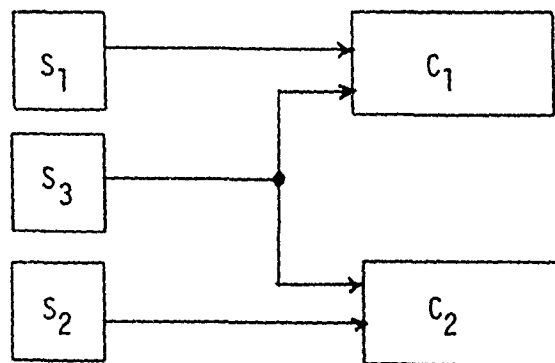
which is \ll the target of 21×10^{-6} . With crossfeeding

$$P[\text{Total Failure}] = .00164 \times 10^{-6}$$

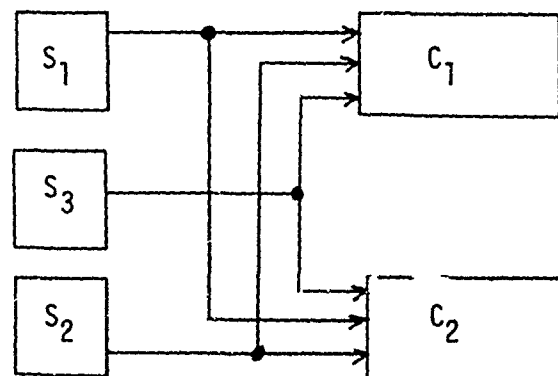
a 33% improvement. Crossfeeding is, again, a benefit, but not a requirement during the recovery phase.

CONFIGURATION 5

For configuration 5 there are two ways to crossfeed three sensors into two computation channels.



Configuration 5a



Configuration 5b

Configuration 5a crossfeeds the third sensor only. However, if the crossfeed sensor fails, both channels are not useable because the remaining sensors are not monitored. The probability of a failure for this configuration is given as:

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P[S_1] P[S_2] + P[S_3] + P[C_1] P[C_2] \\
 &\quad + P[S_1] P[C_2] + P[S_2] P[C_1] \\
 &= 0.0197 \quad \text{per flight}
 \end{aligned}$$

This failure rate is not acceptable.

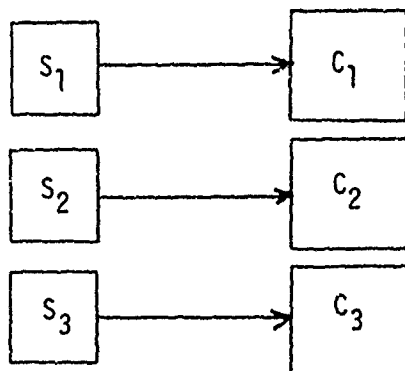
Configuration 5b crossfeeds all three sensors. This eliminates the single sensor failure problem.

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P[S_1] P[S_2] + P[S_1] P[S_3] + P[S_2] P[S_3] \\
 &\quad + P[C_1] P[C_2] \\
 &= 1.13 \times 10^{-3} \quad \text{per flight}
 \end{aligned}$$

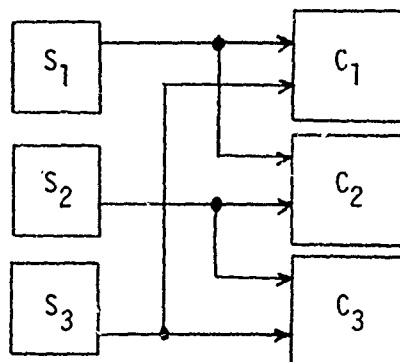
The need for complete sensor crossfeeding in Configuration 5 is obvious from the above numbers.

CONFIGURATION 6

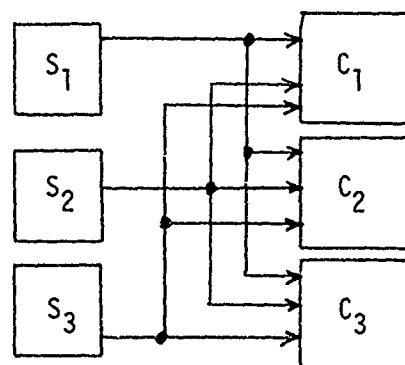
Configuration 6 has three ways that three sensors can be used with three computations. These are shown below:



Configuration 6a



Configuration 6b



Configuration 6c

Configuration 6a is not able to cross compare the sensors. Only the outputs of the computations are cross compared. Sensor failures are only detected by the computation output comparators.

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P[S_1] P[S_2] + P[S_2] P[S_3] + P[S_1] P[S_3] \\
 &\quad + P[C_1] P[C_2] + P[C_1] P[C_3] + P[C_2] P[C_3] \\
 &\quad + P[S_1] P[C_2] + P[S_1] P[C_3] + P[S_2] P[C_1] \\
 &\quad + P[S_2] P[C_3] + P[S_3] P[C_1] + P[S_3] P[C_2]
 \end{aligned}$$

$$= 3 P^2[S] + 3 P^2[C] + 6 P[S] P[C]$$

$$= 1.89 \times 10^{-3} \quad \text{per flight}$$

Configuration 6b uses a limited crossfeed of the sensors, thus certain cross channel failures can be tolerated.

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= P[S_1] P[S_2] + P[S_1] P[S_3] + P[S_2] P[S_3] \\
 &\quad + P[C_1] P[C_2] P[C_3] + P[S_1] P[C_3] \\
 &\quad + P[S_2] P[C_1] + P[S_3] P[C_2] \\
 &= 3 P^2[S] + P^3[C] + 3 P[S] P[C] \\
 &= 1.43 \times 10^{-3} \text{ per flight}
 \end{aligned}$$

Configuration 6c crossfeeds all three sensors to all three computation channels. Thus, all cross channel failures are eliminated.

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= 3 \lambda_s^2 \tau^2 + \lambda_c^3 \tau^3 \\
 &= 1.09 \times 10^{-3} \text{ per flight}
 \end{aligned}$$

Configuration 3b has a 31% higher probability of failure per flight than does Configuration 3c.

Summary

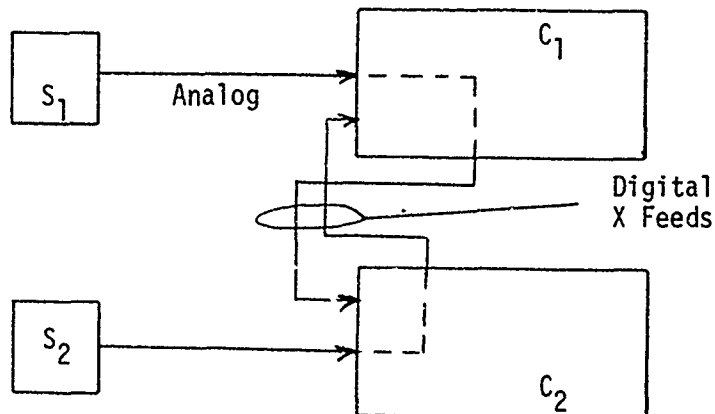
A review of the numbers for the configurations using three unmonitored sensors shows that crossfeeding is necessary in order to keep the failures per flight within acceptable bounds. There is only the case, configuration 2, where the need for sensor crossfeeding is not apparent.

SENSOR INTERFACING

This section addresses the method of crossfeeding sensor data into the flight computers.

Dual Sensors

In general, crossfeeding dual sensors with digital crossfeeds between flight computers is unsafe. It is difficult to process and convert raw sensor data in a flight computer without increasing the potential for undetected failures in the sensor path.

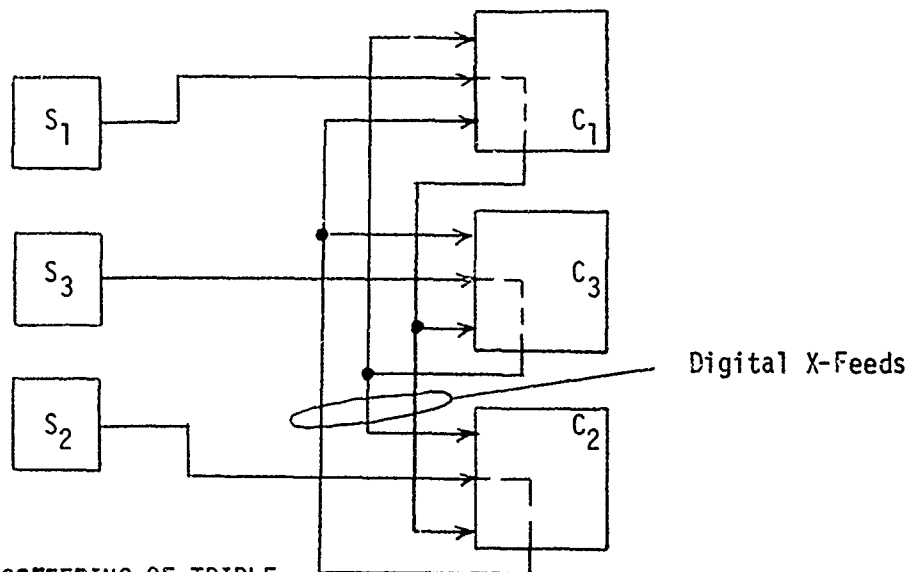


DIGITAL CROSSFEEDING OF DUAL SENSORS

Fortunately the need for crossfeeding in the case of dual sensors feeding dual computers, configuration 2, has not been established. In the case of dual sensors feeding triple computers, configuration 3, symmetry makes digital crossfeeding undesirable.

Triple Sensors

Crossfeeding has been shown to be necessary in the triple-sensor configurations, 5b and 6c. In the case of triple sensors into triple computers, 6c, digital crossfeeding between computers is definitely cost effective. The number of synchro buffers and power normalizers associated with the vertical gyro inputs, for example, can be reduced from 18 to 6 on a system basis.



DIGITAL CROSSFEEDING OF TRIPLE SENSORS

In the case of triple sensors into dual computers the cost advantage of digital crossfeeding is not obvious. The saving in sensor I/O is overcome by the additional complexity of digital crossfeeding circuitry.

Sensor Voting

Once crossfeeding of sensors is established, voting becomes desirable for reasons discussed below under Servo Voting. Sensor voting can be accomplished in software at low cost per voted set.

DISCUSSION ON THE NEED FOR FAIL OPERATIVE SERVOS

The servo (including the servo amplifier) considered for this study has a failure rate of 75×10^{-6} failures/hour. A single servo on a 24 hour flight would have a failure probability of 0.00180 failures/flight.

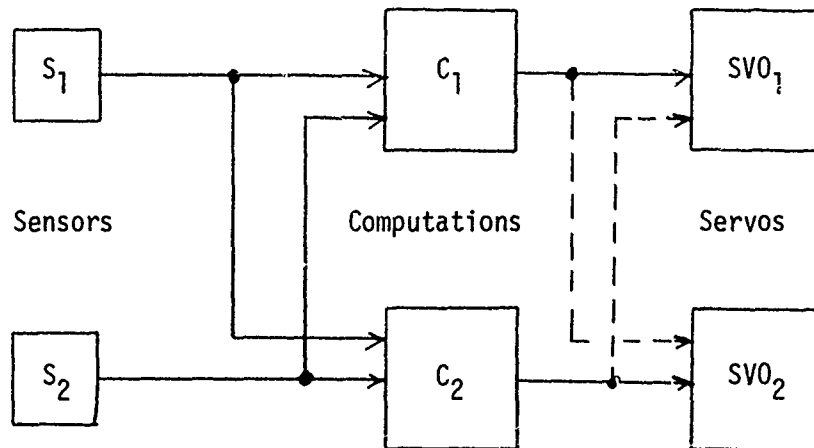
There are four flight-critical control channels: pitch, roll, yaw and power. If each of these channels has only one servo, the probability of a loss due to a servo failure during a flight becomes 0.0072. This probability of failure is too large (the target probability is 0.0017 failures/flight.)

The proposed servo is a fully in-line monitored unit. Using two servos per channel will provide a fail-operative capability. The probability of failure of one flight-critical channel that uses dual servos is 3.24×10^{-6} failures/flight. For the four flight-critical channels this becomes 1.296×10^{-5} failures/flight.

Servo Command Crossfeeding

For the system configuration that has three computation channels to drive the two servos, it is obvious that crossfeeding of the servo commands is necessary to utilize the three computations.

Crossfeeding of dual channel computations into dual servos is not so obvious. Consider the figure below:



Dual Computations and Servos

For this case consider that the sensors, computations and servos are 100% in-line monitored. From the section on The Need for Fail Operative Sensors, use the composite sensor failure rate of $\lambda_s = 7.95 \times 10^{-4}$ failures/hour and the computation channel failure rate of $\lambda_c = 2.5 \times 10^{-4}$ failures/hour. As stated in the previous section, the failure rate of the servo is $\lambda_{sv0} = 75 \times 10^{-6}$ failures/hour. The flight time per mission is 24 hours.

If the servo commands are not crossfed, the total system failure rate can be calculated to be:

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= (\lambda_s^2 \tau^2 + \lambda_c^2 \tau^2 + \lambda_{sv0}^2 \tau^2) + 2\lambda_c \lambda_{sv0} \tau^2 \\
 &= 4.72 \times 10^{-4} \quad \text{per flight}
 \end{aligned}$$

Crossfeeding the servo commands (the dashed lines in the figure) eliminates the cross channel failures.

$$\begin{aligned}
 P[\text{TOTAL FAILURE}] &= \lambda_s^2 \tau^2 + \lambda_c^2 \tau^2 + \lambda_{sv0}^2 \tau^2 \\
 &= 4 \times 10^{-4} \quad \text{per flight}
 \end{aligned}$$

There is only a 14% improvement in the failures per flight as a result of crossfeeding the servo commands. The need for crossfeeding the servo commands is not necessary from the failure rate analysis. However, other considerations discussed elsewhere may show a need for crossfeeding the servo command in this configuration.

Servo Interfacing - Command Switching vs. Voting

Triple Computers

Two techniques can provide tracking autopilot commands to the downstream servos - switching and voting, as shown in Fig. A-1. At a modest cost penalty, triplex analog voters can provide superior hardware rejection and a lower nuisance disconnect rate. Voters do not require the cross-channel comparators to reject hardovers, but only to reconfigure the voters following the first failure. Consequently, the comparators can operate significantly slower than in the command-switching scheme.

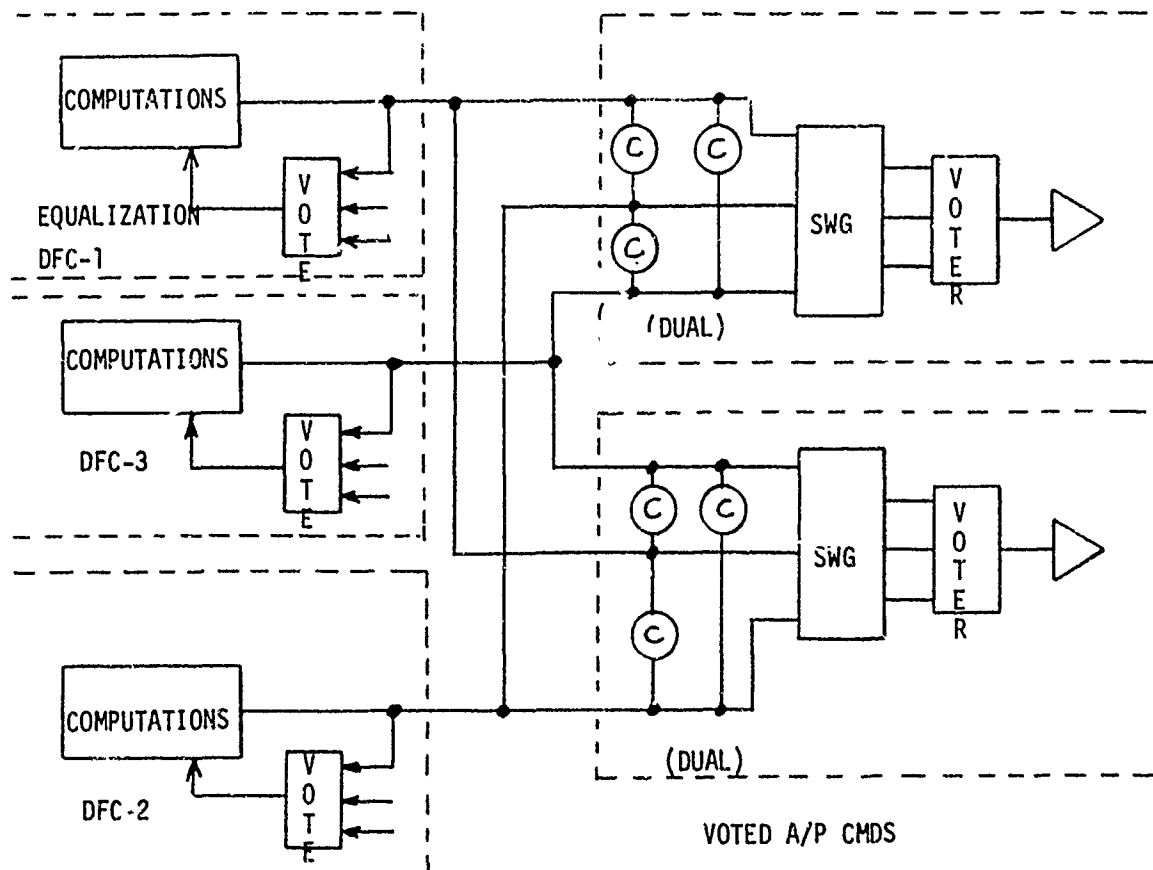
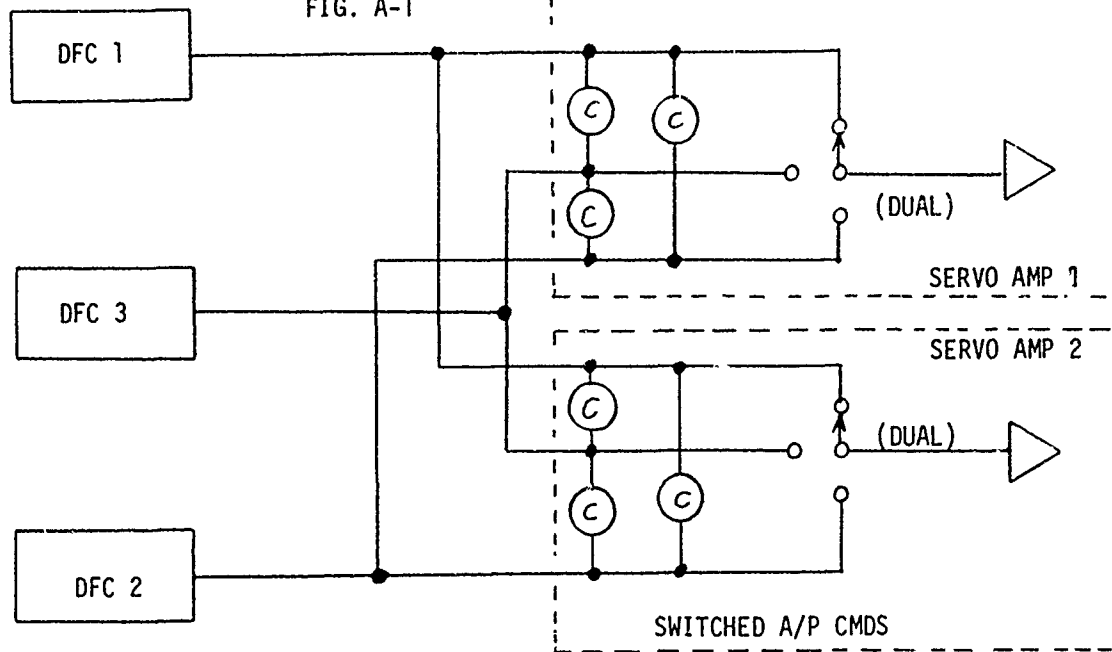
Since the computations implemented within the DFC's will contain forward-path integrators in some modes, equalization must be provided to preclude autopilot command divergence. An equalization signal can be generated by voting the triple autopilot commands in each DFC, Fig. A-1. This voting can be performed in software. The same algorithm used for sensor voting can be used for equalization.

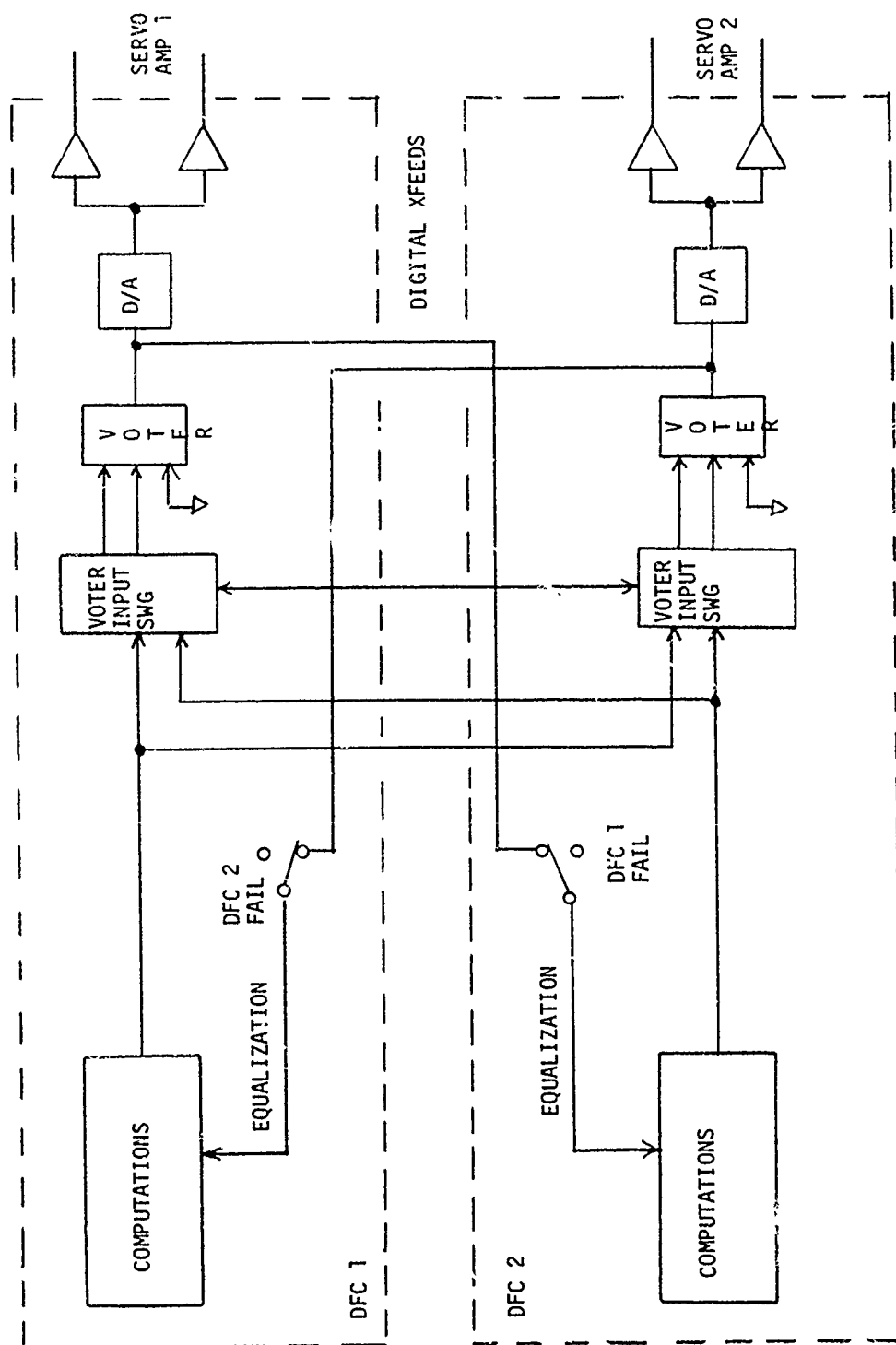
Dual Computers

Autopilot command crossfeeding is not required from a fault analysis standpoint, as discussed above. Depending on the servo configuration, however, the commands must be forced to track. Additionally, equalization is required. Since software output voters can be added to the DFC's at negligible cost, both improved command tracking and an equalization means can be obtained, Fig. A-2. The equalization feed must be broken, appropriately, to avoid cross-contamination when one DFC fails. It is recognized that, in some cases, cross-contamination cannot be avoided. However, this event can be handled like an undetected hardware failure and can be caught with redline monitoring.

SERVO INTERFACE COMMAND SWITCHING VS. VOTING, TRIPLE COMPUTERS

FIG. A-1





SERVO INTERFACE
DUAL COMPUTERS

FIG. A.2

APPENDIX B

DETAILED FAULT ANALYSIS

The detailed fault trees used to generate the various vehicle-loss probabilities for Configurations B, C, D, and E are included in this appendix. Configuration A is just a subset of Configuration B. The fault trees are self explanatory and include the fault probabilities for the various branches. A definition of mnemonics is included.

The example started in Section 5 for vehicle loss in recovery is discussed in detail for Configuration D.

APPENDIX B
DEFINITION OF MNEMONICS

A/P	Autopilot
DG	Directional Gyro (Compass System)
D/L	Data Link
FCILC	Flight Critical Inner Loop Control
FGC	Flight Guidance Computations (Same as Digital Flight Computer for configurations chosen)
FGCD	Detected Failures of the FGC
FGCU	Undetected Failures of the FGC
GA	Go-Around
h_c	Critical Altitude (50 ft)
MLS	Microwave Landing System Receiver
R/L	Redline Monitor
RA	Radio Altimeter
REC	Recovery Phase
RO	Remote Operator
SOFT	Software
SVO	Servo
T/O	Takeoff Phase
VG	Vertical Gyro

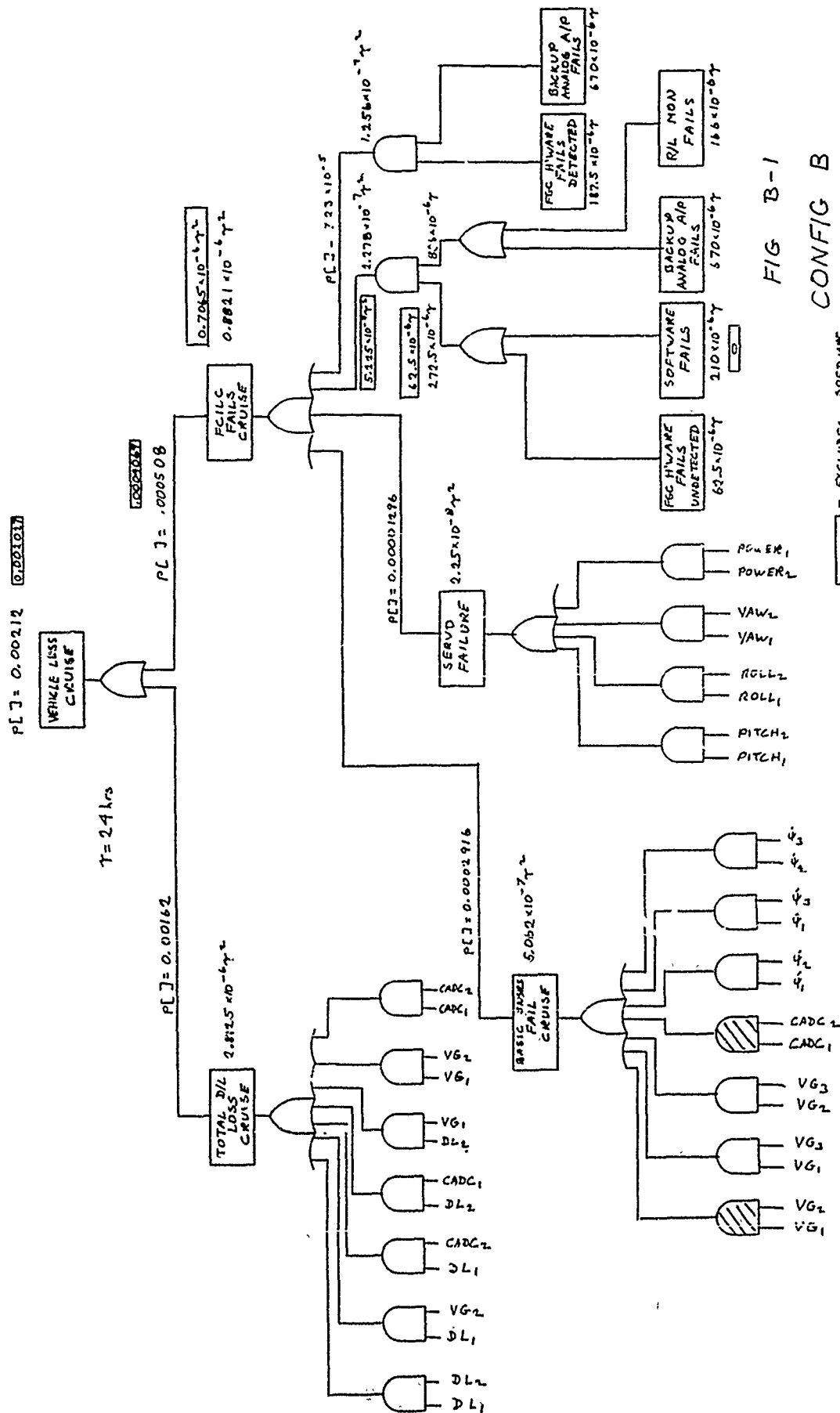


FIG B-1

CONFIG B VEHICLE LOSS, CRUISE

21 NOV 75

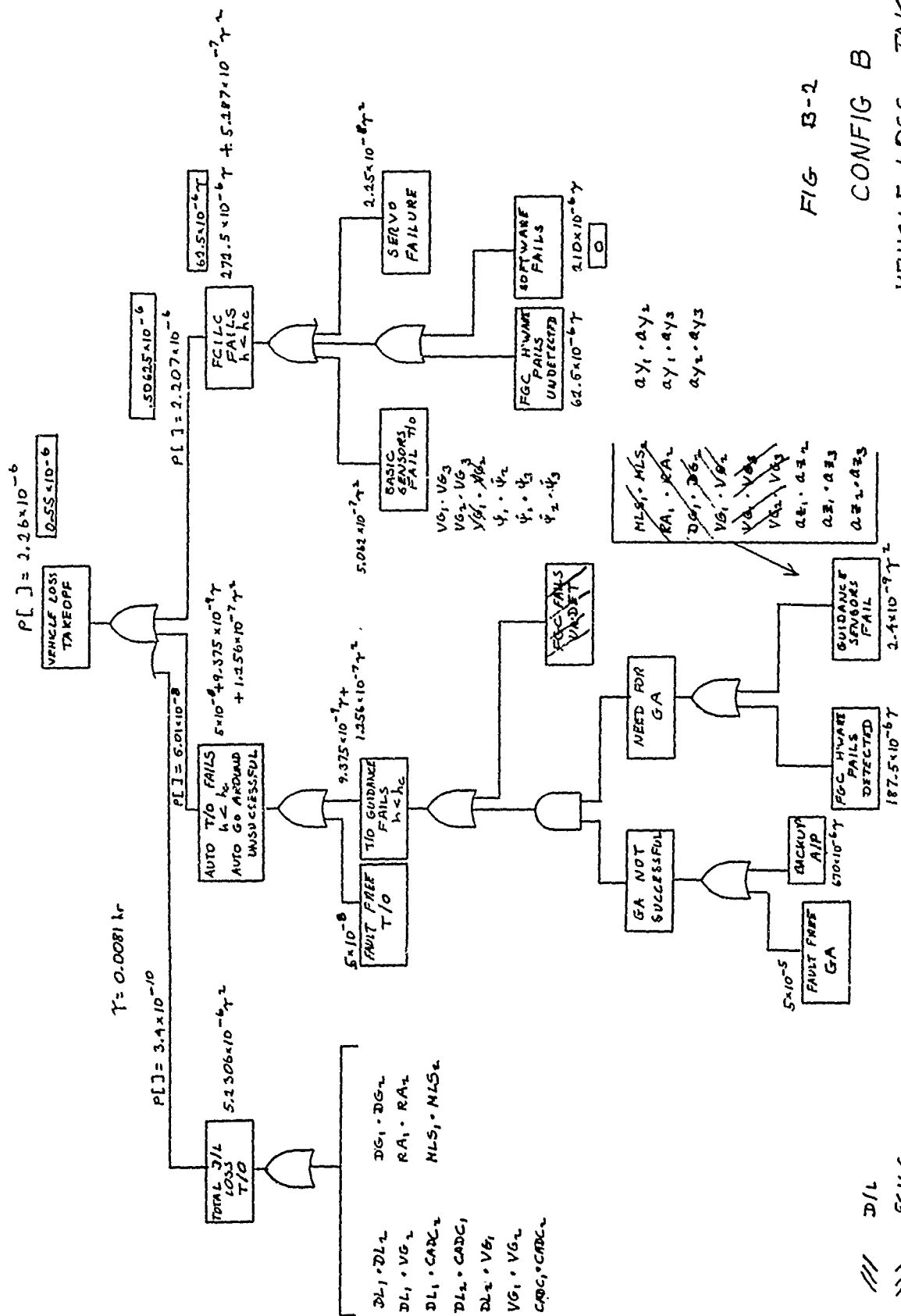


FIG B-2

CONFIG B

VEHICLE LOSS, TAKEOFF

REF 14 NOV 75

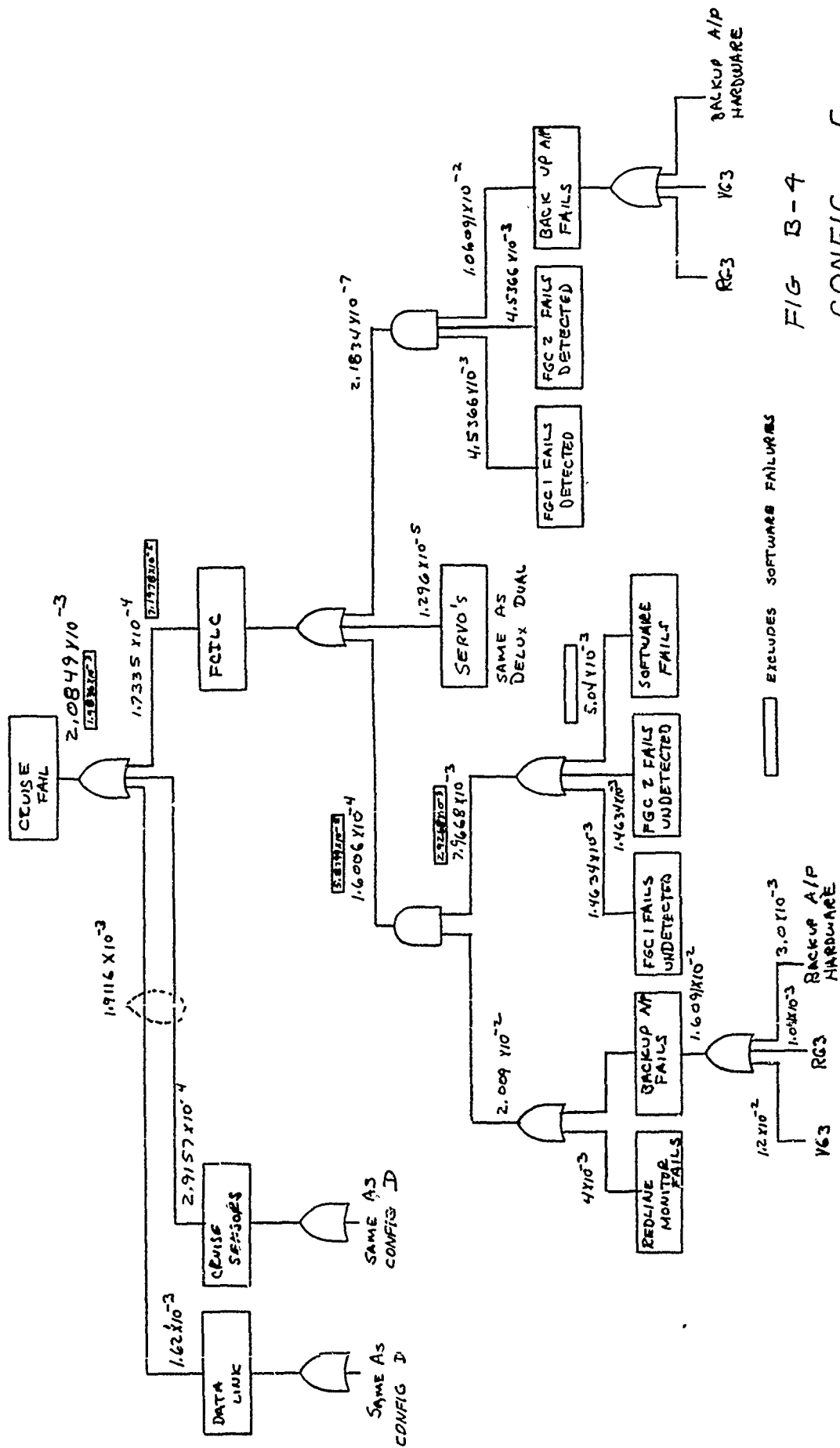
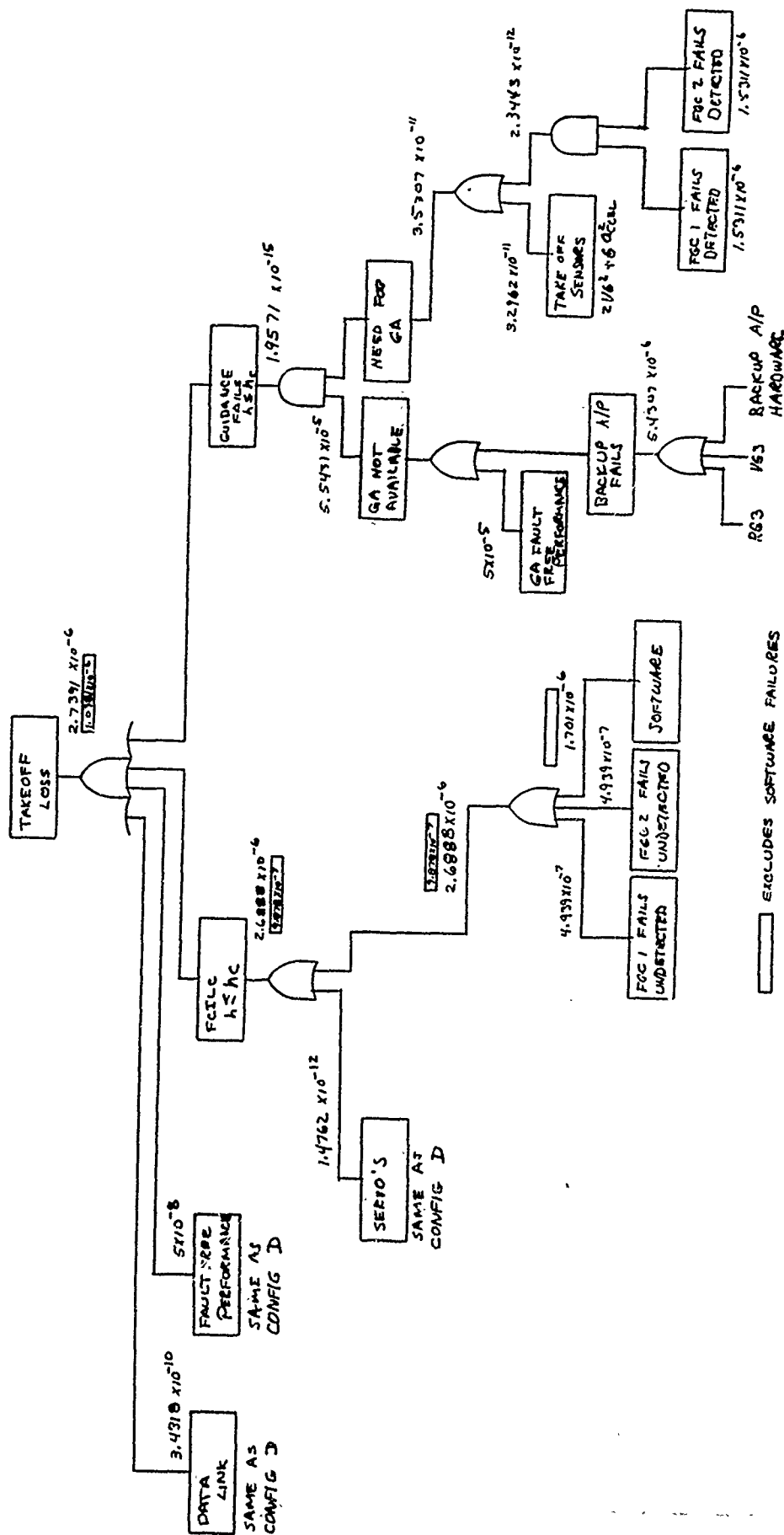


FIG B-4

CONFIG C
VEHICLE LOSS, CRUISE



CONFIG C
VEHICLE LOSS, TAKEOFF
FIG B-5

APPENDIX B
DETAILED FAULT ANALYSIS OF
CONFIGURATION D,
RECOVERY PHASE

The fault trees are generated from the top down. The failure probabilities are then calculated from the bottom up.

$$P[\text{RECOVERY LOSS}] = P[\text{D/L LOSS}] + P[\text{FAULT FREE PERFORMANCE}] \\ + P[\text{FCILC FAILS}] + P[\text{GUIDANCE FAILS}]$$

The data link loss includes not only the data link, but the sensors necessary for the remote operator to assess the safe operation of the vehicle. The hazardous faults for $P[\text{D/L Loss}]$ are listed in the following equation:

$$P[\text{D/L LOSS}] = P[\text{DG 1}] P[\text{DG 2}] + P[\text{D/L 1}] P[\text{DG 2}] + P[\text{D/L 2}] P[\text{DG 1}] \\ + P[\text{RA 1}] P[\text{RA 2}] + P[\text{D/L 1}] P[\text{RA 2}] + P[\text{D/L 2}] P[\text{RA 1}] \\ + P[\text{MLS 1}] P[\text{MLS 2}] + P[\text{D/L 1}] P[\text{MLS 2}] + P[\text{D/L 2}] P[\text{MLS 1}] \\ + P[\text{VG 1}] P[\text{VG 2}] + P[\text{D/L 1}] P[\text{VG 2}] + P[\text{D/L 2}] P[\text{VG 1}] \\ + P[\text{D/L 1}] P[\text{D/L 2}]$$

Since all No. 1 units are the same as the No. 2 units, the above equation can be rewritten:

$$P[\text{D/L LOSS}] = P^2[\text{DG}] + 2 P[\text{D/L}] P[\text{DG}] \\ + P^2[\text{RA}] + 2 P[\text{D/L}] P[\text{RA}] \\ + P^2[\text{MLS}] + 2 P[\text{D/L}] P[\text{MLS}] \\ + P^2[\text{VG}] + 2 P[\text{D/L}] P[\text{VG}] \\ + P^2[\text{D/L}]$$

However,

$$P[A] = \lambda_A T$$

where λ_A is the failure rate and T is the exposure time. Thus the equation becomes:

$$P[D/L \text{ LOSS}] = (\lambda_{DG}^2 + 2\lambda_{DG}\lambda_{DL} + \lambda_{RA}^2 + 2\lambda_{RA}\lambda_{DL} + \lambda_{MLS}^2 + 2\lambda_{MLS}\lambda_{DL} + \lambda_{VG}^2 + 2\lambda_{VG}\lambda_{DL} + \lambda_{DL}^2) T^2$$

Using the values from the Table 8-1 in Section 8, this equation becomes:

$$P[D/L \text{ LOSS}] = 5.23 \times 10^{-6} T^2$$

The recovery phase from 1500 feet altitude to the end of the rollout takes 0.0409 hours (See Section 4). Using this exposure time the probability of vehicle loss due to a data link loss is

$$P[D/L \text{ LOSS}] = 8.75 \times 10^{-9} \text{ per flight}$$

The fault free performance value used for this study was 5×10^{-8} failures per flight.

The $P[\text{Recovery Loss}]$ due to FCILC failures is comprised of those failures that occur below h_c and those that occur above h_c . If any two servos fail in any one of the four flight-critical controls, a vehicle loss occurs above or below h_c .

Below h_c if both computers fail detected, both internal microprocessors must also fail before a loss occurs, since either computer can perform a go-around. If either computer fails undetected, a loss occurs. If a software failure is considered, it is common to both computers and then a single failure causes a loss.

Putting this into equation form with $T = 0.0068$ gives

$$\begin{aligned} P[FCILC \ h < h_c] &= 4\lambda_{svd}^2 T^2 + 2\lambda_{FGCU} T + \lambda_{PP}^2 \lambda_{FGCD}^2 T^4 \\ &\quad + \lambda_{SOFT} T \\ &= 1.58 \times 10^{-6} \text{ per flight} \end{aligned}$$

Without software failures this becomes

$$P[FCILC \ h < h_c] = 1.55 \times 10^{-7} \text{ Failure/Flight (No Software Failures)}$$

Other than the software failure, the next largest factor in this probability is the unmonitored DFC failures.

Above h_c the redline monitor can detect the unmonitored DFC hardware failures and the software failures, give warning to the remote operator, and the RO can then assume control. The redline monitor cannot help in the event of servo failures. Above h_c the equation becomes

$$\begin{aligned} P[FCILC \ h > h_c] &= 4\lambda_{SVO}^2 T^2 + \lambda_{PP}^2 T^2 (\lambda_{FGCD}^2 T^2 \\ &\quad + 2\lambda_{FGCD} T + \lambda_{SOFT} T) \\ &= 2.62 \times 10^{-11} \text{ per flight} \end{aligned}$$

Note that with the redline monitor only the servo failures contribute to any significant failure rate.

Guidance failures are also separated into those below h_c and above h_c . Above h_c the remote operator can assume control and below h_c a go-around can be initiated for a detected failure. The equation for this can be written from the diagram in the same manner as for the FCILC failures.

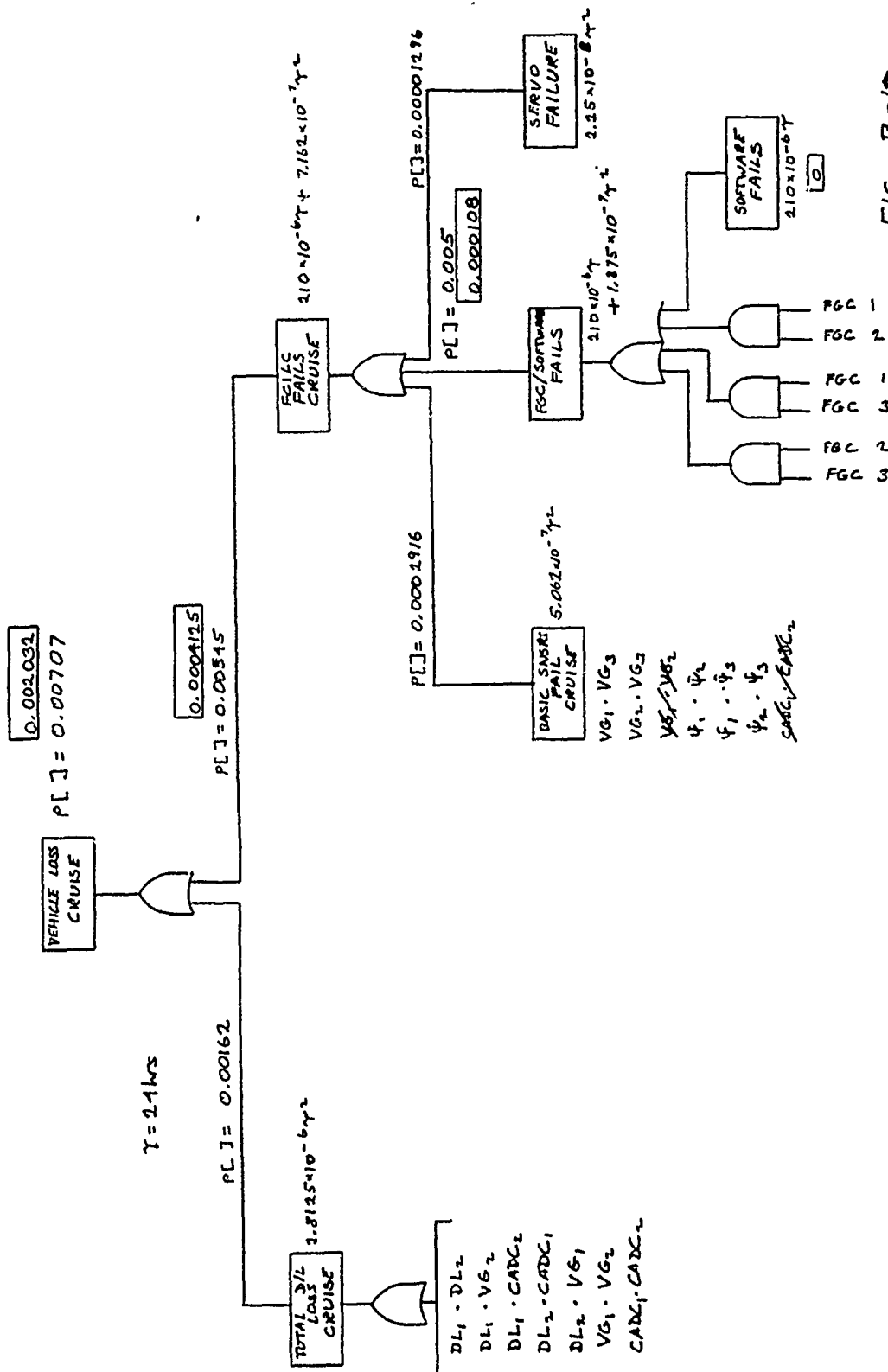
$$\begin{aligned} P[\text{Guidance Fails}] &= P[\text{Guidance } h < h_c] + P[\text{Guidance } h > h_c] \\ &= 1.69 \times 10^{-10} \text{ per flight} \end{aligned}$$

Combining all of these terms neglecting software failures gives:

$$\begin{aligned} P[\text{Recovery Loss}] &= P[D/L \text{ Loss}] + P[\text{Fault Free Performance}] \\ &\quad + P[FCILC \text{ Fails}] + P[\text{Guidance Fails}] \\ &= 8.75 \times 10^{-9} + 5 \times 10^{-8} + (1.55 \times 10^{-7} + 2.62 \times 10^{-11}) \\ &\quad + (2.31 \times 10^{-11} + 1.46 \times 10^{-10}) \\ &= 2.14 \times 10^{-7} \text{ per flight} \end{aligned}$$

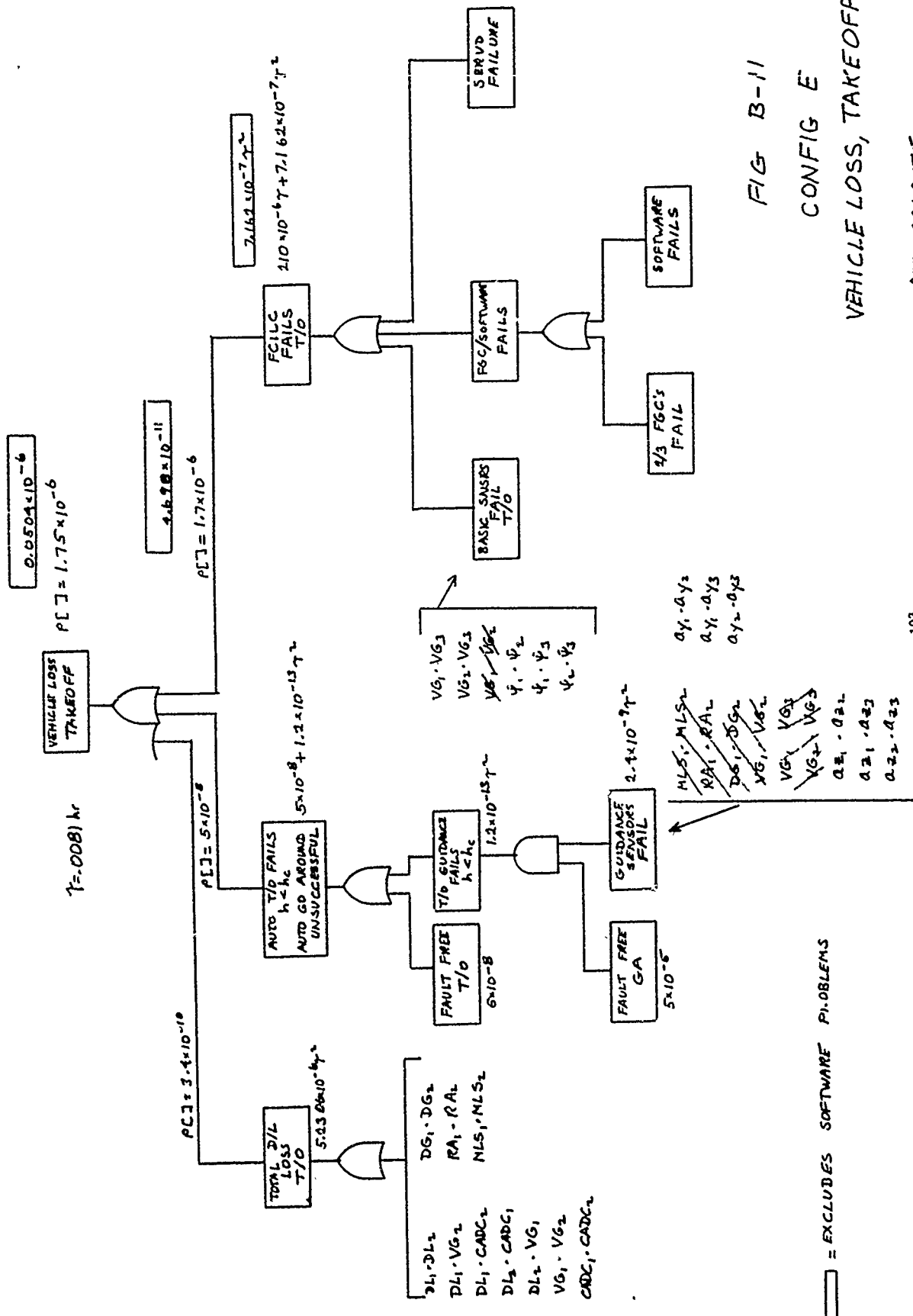
It is interesting to note that the biggest factors in this result are the control low fault free performance and the FCICC failures below h_c .

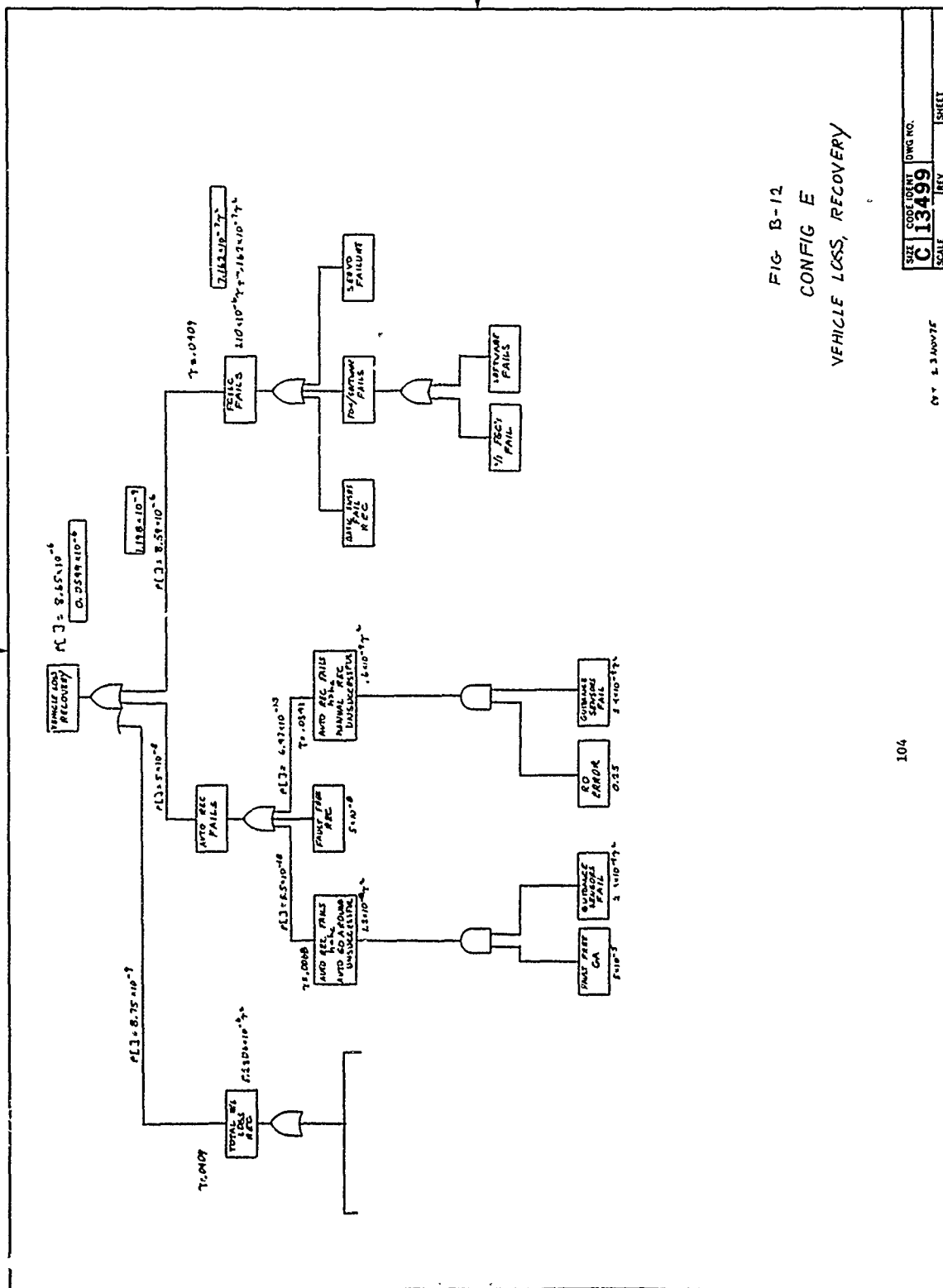
If the software failure rate is included the recovery loss is about 8 times worse or about 1.64×10^{-6} failures/flight.



□ = EXCLUDES SOFTWARE FAILURES

FIG B-10
 CONFIG E
 VEHICLE LOSS, CRUISE





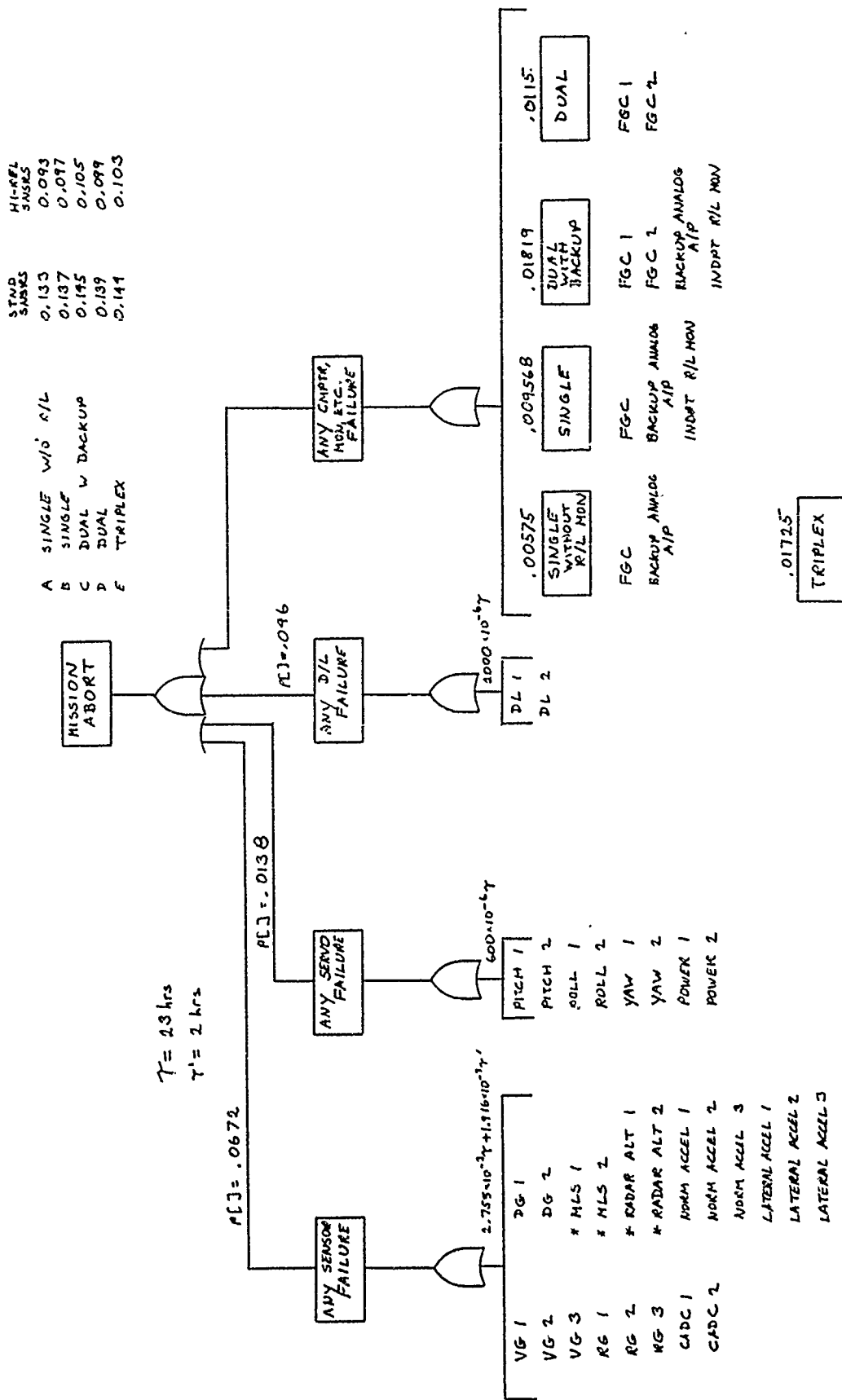


FIG B-13
MISSION ABORT

FGC 1
FGC 2
FGC 3

NOTE
* .SENSORS UNPOWERED ABOVE 10,000 FT NSL ($T' = 2 \text{ HRS}$)
 ** WITH HI-REL VG (350×10^{-6}) + D/L (400×10^{-6}) REDUCE TOTALS BY .04025.

APPENDIX C

DERIVATION OF NON-RECURRING

PLANNING ESTIMATES

As requested by the Statement of Work, the one-time costs associated with the hardware design and system integration are derived below for the five configuration candidates. These costs are estimates and should be used only for planning and relative comparison of candidate configurations.

The configuration candidates defined and evaluated in the body of the report were built from an equipment list of both off-the-shelf and new, as yet undesigned, equipment. The non-recurring totals for each candidate, therefore, include new equipment design and production start-up costs and system engineering costs. The system engineering task includes sensor/servo selection, equipment integration, and program management. Control-law development and detailed failure mode and effect analysis, as required for Category III autoland certification, are not included. Similarly, data preparation is omitted.

System Engineering Estimate

Sensor Selection	57K
Servo Selection	57K
System Integration	115K
Program Management	<u>57K</u>
	\$286K

New Equipment Design Estimates

Conventionally-Monitored DFC	
Hardware Development	255K
Software Development	174K
Misc. Expense*	<u>153K</u>
	\$582K

Highly-Monitored DFC	
Hardware Development	277K
Software Development	189K
Misc. Expense*	<u>166K</u>
	\$632K

Analog Flight Computer	
Hardware Development	232K
Misc. Expense*	<u>139K</u>
	\$371K

7-2-1

APPENDIX C (Continued)

New Equipment (Continued)

Independent Red-line Monitor	
Hardware Development	232K
Software Development	196K
Misc. Expense*	<u>139K</u>
	\$567K
 Servo Amplifier	
Hardware Development	56K
Misc. Expense*	<u>34K</u>
	\$ 90K

*Includes production collateral support, qual. test, reliability and maintainability expense. Does not include data.

APPENDIX C (Continued)

Total Non-Recurring Estimates by Configuration

Configuration A	
Servo Amplifier Development	90K
Analog Flight Computer Development	371K
Conventionally-Monitored DFC Development	582K
System Engineering	<u>286K</u>
	\$1329K
Configuration B	
Configuration A Non-Recurring Costs	1329K
Independent Red-line Monitor Development	<u>567K</u>
	\$1896K
Configuration C	
Same as Configuration B	\$1896K
Configuration D	
Servo Amplifier Development	90K
Highly-Monitored DFC Development	632K
System Engineering	<u>286K</u>
	\$1008K
Configuration E	
Servo Amplifier Development	90K
Conventionally-Monitored DFC Development	582K
System Engineering	<u>286K</u>
	\$ 958K

APPENDIX D
AIRCRAFT LOSS DATA¹
1973

U. S. AIR CARRIERS, ALL OPERATIONS

Hours Flown	6.5×10^6
Total Losses	7
Landing Losses	5
Takeoff Losses	1
Loss Rate	$1.076 \times 10^{-6} \text{ hr}^{-1}$
Fraction Landing Losses	0.714
Fraction Takeoff Losses	0.143

¹References 11 and 12.

APPENDIX D (Continued)

GENERAL AVIATION, ALL OPERATIONS

Hours Flown	30×10^6
Total Losses	1102
Landing Losses	256
Takeoff Losses	169
Loss Rate	$36.7 \times 10^{-6} \text{ hr}^{-1}$
Fraction Landing Losses	0.232
Fraction Takeoff Losses	0.153